Discussion and Modifications:

Throughout the milestone a few changes were made:

1. First I created a 'move' object that stores and keeps track of all the move options that a player can make from the board. Move stores al the played regular tiles , special tiles, their locations on board, a flag for when a pass is made, and if the move only consists of purchasing a special tile.

2. Then the move is played using the Play method in the Scrabble class. Play first checks if the move is valid calling a bunch of test methods. If it is then the move is made, board is updated and score is calculated and added to player's current score.

3. The name of 'getSpecial' was changed to 'getSpecialTile' for purchasing new tiles.

4. The class responsible for figuring out which words are valid was changed from scrabble to "Board". This is because at any point board holds the current state of tiles held on it during the game. This prevents any unnecessary interaction between the classes and made it much simple to code for the process. The score for a valid move was also calculated in the same class.

5. Added a check for the rule that all tiles in a move must be collinear (Horizontal or Vertical) that was not being checked earlier.

6. Added a check that there is no empty spot between the head and tail of all placed tiles (i.e. all placed tiles are connected).

7. Created a check that at least one placed tile is adjacent to an existing tile on board (or at the center if no tile currently exists on board i.e. start state).

8. Added a score multiplier attribute to the Scrabbleclass that takes care of the revised scoring for the player when one of the special tiles is activated.

   Apart from these changes there were other changes made and modifications done once I started with the implementations. There were a lot of helper functions added and a few more methods were also added to facilitate communication between classes. At some places the interaction diagrams were revised as things got more clear during implementation.

   Al the special tiles were implemented the same way. Once it was activated it took in the entire game state as its argument . Since the whole game was passed on it was easy to make changes by using a similar state path for all the special tiles. All tiles were handled similarly except for their instructions in the effect method and they were called from the same state in the game.