Johnathon Hein

CM301

CSSE375

Milestone 2 refactoring

# Changing the Piece color to be an Enum rather than Boolean.

*Trello:* [*https://trello.com/c/BtPrg9wm/18-refactor-remove-unused-method-in-piece-class-and-associated-classes*](https://trello.com/c/BtPrg9wm/18-refactor-remove-unused-method-in-piece-class-and-associated-classes)

*Merge request:* [*https://ada.csse.rose-hulman.edu/zhangq2/Chess/merge\_requests/9*](https://ada.csse.rose-hulman.edu/zhangq2/Chess/merge_requests/9)

## Bad code smells And Changes:

The

# Chess.getReachableSquares() refactoring:

*Trello:* [*https://trello.com/c/BtPrg9wm/18-refactor-remove-unused-method-in-piece-class-and-associated-classes*](https://trello.com/c/BtPrg9wm/18-refactor-remove-unused-method-in-piece-class-and-associated-classes)

*Merge request:* [*https://ada.csse.rose-hulman.edu/zhangq2/Chess/merge\_requests/14/*](https://ada.csse.rose-hulman.edu/zhangq2/Chess/merge_requests/14/)

## Bad code smells And Changes:

The original Chess object has many functions that make it seem like a horrible God class. In the process of refactoring the Piece class, I noticed one of these functions that can easily moved into the Piece class. This was chess.getReachableSquares().

This function clearly is using the Piece’s data more than the Chess objects data. It simply works off its parameter, a Piece object. This functionality could just as easily be in the Piece class.

In the end I made two primary changes for this refactor. The first was to move the function into the Piece class and update references to it. Then I moved on to testing the function to verify its functionality. This was definitely the hardest part of the change. The function relies on many different dependencies (Chess, Square, Board, etc.). I used the mocking framework a ton to test the function for its desired behavior. Luckily, the code had fully used dependency injection, so the tests were not impossible.