Chess Assignment

Mahmoud Abdallah

Introduction:

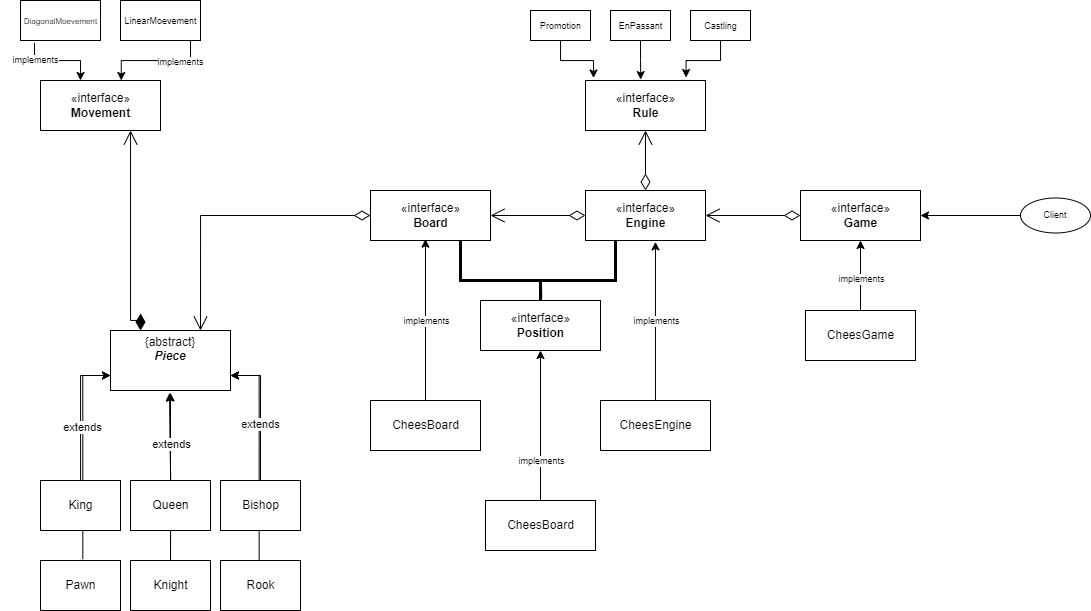
In this assignment we are asked to design and implement a chess game, using OOP and design patterns considering clean code and SOLID principles.

OOP & Design Patterns:

I have created a design which can serve any board game and used it to create the chess game.

The design uses dependency Injection, and Strategy Design Pattern for injecting the rules to the classes that meant to use it, which both decrease coupling between the classes, reduce redundancy and follows the SOLID and Clean Code principles.

The design consists of seven classes/interfaces as follows:



Clean Code & SOLID principles:

Following The Dependency Inversion and the Open-Closed Principles, all classes rely on interfaces and abstractions, which helps to make the design loosely coupled and open for extension.

Regarding The Single Responsibility Principle, the chess game has been divided into two classes (***Engine*** and ***Game***), the ***Engine*** is only responsible for running the game and shouldn’t care about representing the game, and ***Game*** is only responsible for about representing the game and has no clue how it works.

Since all subclasses follow their Super Class, and all relations are based on abstraction, the design achieves the Liskov Substitution and Interface Segregation Principles.

Using the Strategy design pattern reduced the reliability on conditional statements and increased cohesion, I tried to use the same conventions in declaring variables and implementing methods to improve clarity and reduce the number of method’s arguments as possible.

Test Cases:

There are 3 possible outputs which represent different status.

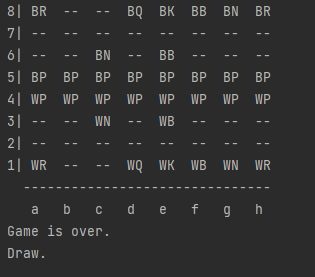
1. Invalid move

Text

Description automatically generatedWhen a player enters an invalid move, the application will show an invalid message and ask them to enter a new message.

Invalid Move

1. Draw

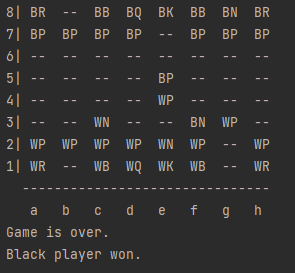
There is a number of moves for a single game that cannot be exceeded. if so, the game will output a draw message

Draw

1. winner

If a checkmate occurs and the there is no move possible for the white king or white pieces, then the game will output a win message and declare the winner.

A screenshot of a computer

Description automatically generated with medium confidence

White Winner - White Queen checks Black King

Black Winner – Black Knight checks White King

References:

1. Clean Code:

<https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29>

1. The SOLID Principles of Object-Oriented Programming Explained in Plain English:

<https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/#:~:text=The%20SOLID%20Principles%20are%20five,and%20software%20architecture%20in%20general>.

1. Defending your code against dependency problems:

<https://software.ac.uk/resources/guides/defending-your-code-against-dependency-problems>

1. 10 Fastest Checkmates

<https://www.chess.com/article/view/fastest-chess-checkmates>