Projektowanie Obiektowe

Krzysztof Król, grupa 1

Projekt pt. **“Chess Solver”** ma na celu zaimplementowanie funkcjonalności odnajdywania ruchu kończącego daną sytuację w partii gry w szachy matem lub patem.

Ma on wykorzystywać przynajmniej 3 wzorce projektowe.

Wykorzystane zostały **4** wzorce:

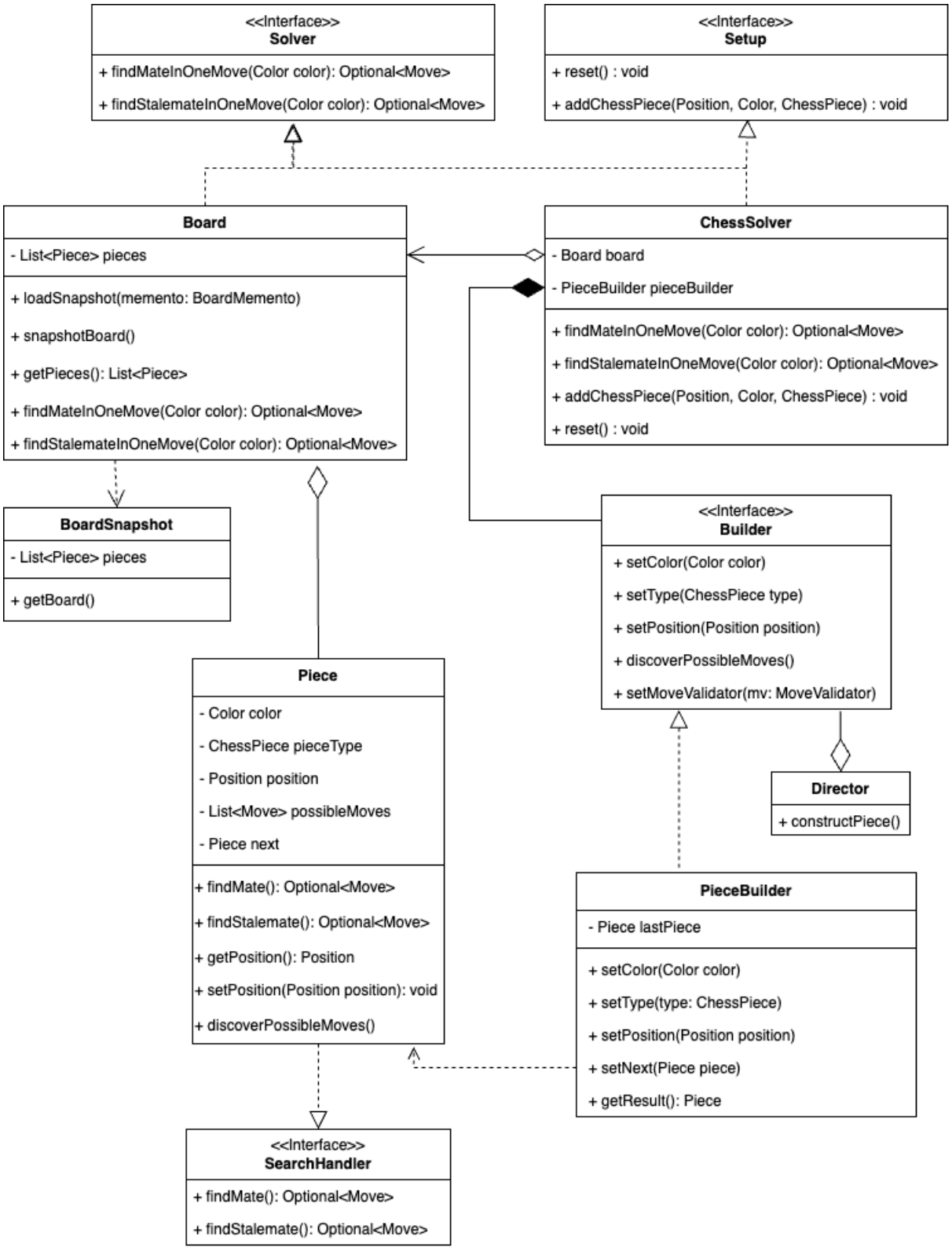
- *Budowniczy* - pozwala na kilkuetapowe wytworzenie odpowiedniego rodzaju figury (**Piece**), ustawiając kolejno jej pola bez potrzeby użycia skomplikowanego konstruktora lub osobnej klasy dla każdego rodzaju figury.

- *Pamiątka* - umożliwia zapisanie stanu w jakiej jest plansza (**Board**) przed ruszeniem się figury i załadowanie go z powrotem np. gdy chcemy sprawdzić możliwość zaistnienia sytuacji pata. Jeśli przeciwnik może się obronić - cofamy plansze do stanu sprzed ruchu.

- *Łańcuch zobowiązań* - każda figura otrzymuję od budowniczego informację o następnej, co pozwala na automatyczne, kaskadowe sprawdzenie możliwości pata lub mata za pomocą jednego wywołania. To rozwiązanie również pozwala zwiększyć wydajność programu, ponieważ algorytm kończy pracę od razu po znalezieniu pasującego rozwiązania. Nie musi dokończyć iteracji po reszcie figur.

- *Pełnomocnik* - klasa **ChessSolver** jest pełnomocnikiem klasy **Board**, nie dopuszczamy użytkownika programu do bezpośredniej interakcji z planszą. Może to pomóc w zapewnieniu wierzytelności wyniku, ponieważ nikt nie może bezpośrednio ingerować w stan planszy np. w trakcie szukania wyniku ładować stan planszy z pamiątki (**BoardSnapshot**). Ponadto, znajomość detali implementacyjnych klasy **Board** nie są potrzebne klasie **ChessSolver**.

Diagram UML:*



*pominięte zostały na nim obiekty typu **enum** z dostarczonego kodu źródłowego oraz klasa **ChessPieceAsUnicode**