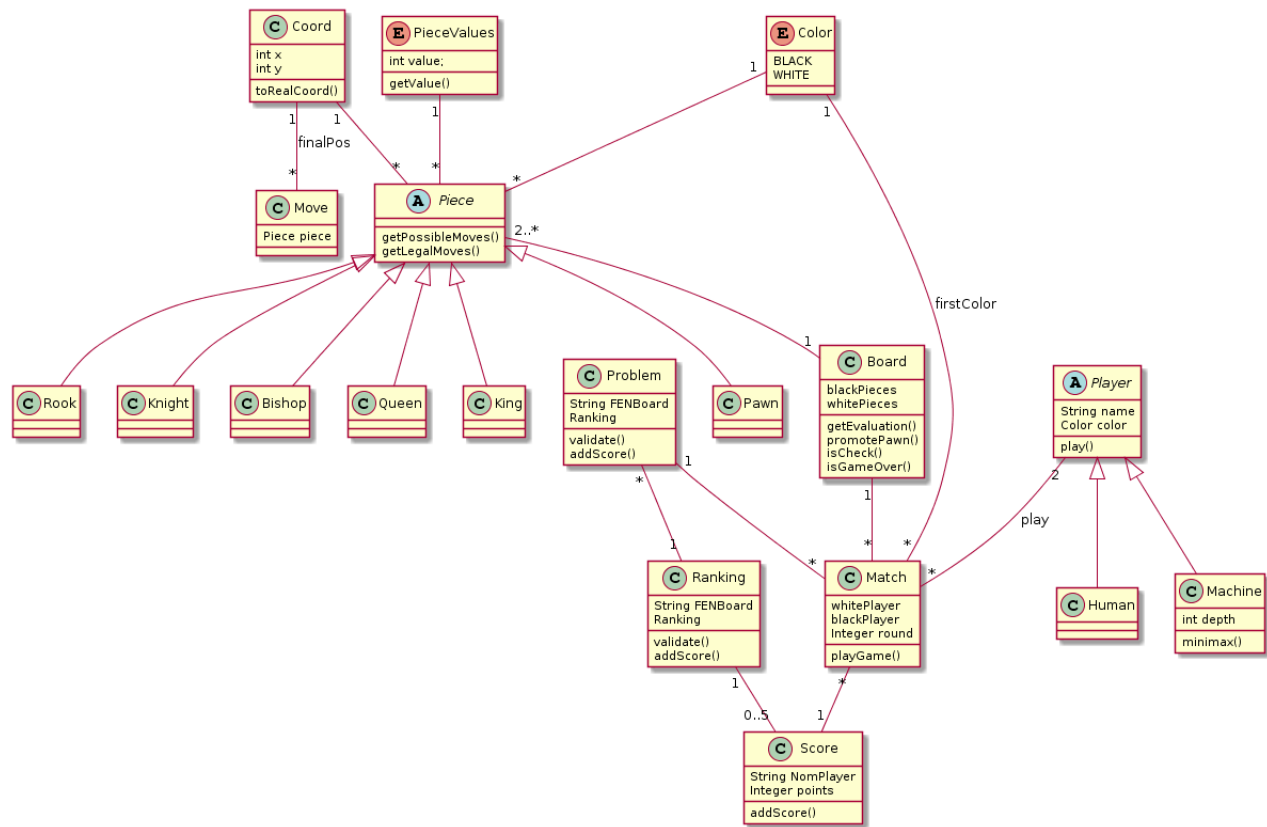


CHESS GAME

May 3, 2019

1 Diagrama UML



- Color: Color és una enumeració que s'usarà com atribut per determinar un equip. Conté els valors "WHITE" o "BLACK".
- Coord: Coord és una estructura formada per dos enters que determinen la posició en el taulell de joc.

- PieceValues: PieceValues és una enumeració que determina quin valor té cada peça dins del joc. S'usa tant per calcular la puntuació del guanyador com perquè els algorismes calculin els millors moviments.
- Piece: La classe Piece conté tota la informació de cada una de les peces del joc. Conté la posició, determinada per la classe Coord, l'equip, determinat per l'enumeració Color, i el seu valor, que prové de l'enumeració PieceValues. Aquesta classe s'estén en 6 classes filles, una per cada tipus de peça que existeix als escacs.
- Move: La classe Move està formada per una posició, continguda dins d'una Coord, i serveix per moure les peces. El moviment el determina la posició de l'atribut i la posició de la peça que es vol moure.
- Board: La classe Board està formada per un conjunt de peces, segons el problema, amb dos ArrayList. Cada ArrayList conté les peces de cada equip.
- Match: La classe Match està formada per un Board, un Color que determina a quin jugador li toca jugar, dos Players i una Score, que guardarà la puntuació del guanyador, i el Problem a solucionar.
- Player: La classe Player és la que conté el nom en forma de String i amb quin equip juga, determinat amb l'enumeració Color. Segons el tipus de jugador Humà o màquina es divideix en dues subclasses: Human: que llegeix els moviments de la terminal; Machine: que conté l'algorisme per trobar els millors moviments per guanyar.
- Score: La classe Score guarda el nom del jugador guanyador del Match i la seva puntuació.
- Ranking: La classe ranking està formada per les 5 millors Score d'un Problem en concret. Quan arriba una Score nova i és superior a qualsevol de les que hi ha a Ranking, la més petita guardada s'elimina i es posa la Score nova a la posició que li correspon.
- Problem: La classe Problem està formada per un FEN, la dificultat del problema, qui comença jugant i dues ArrayList amb la posició de les peces de cada equip així com la quantitat de torns per fer el mat (N).

2 Diagrama de Casos d'ús

‘Per la grandària de la imatge, no hem afegit el diagrama en aquest document. El podeu trobar en format pdf al directori on es troba aquest arxiu. L'usuari disposarà de tres opcions principals: "Game", "Problem Managament" i "Ranking Managament".

2.1 Game

- Create Player: L'usuari podrà crear un nou jugador.
- Load existing Player L'usuari podrà buscar el seu jugador a la base per jugar amb el seu Player.
- Load Game: L'usuari podrà reprendre una partida guardada.
- New Game: L'usuari tindrà l'opció de crear una nova partida.
- Load Problem: Per crear una partida, es llistarà tots els problemes disponibles al sistema per tal que l'usuari pugui triar quin problema jugar.
- Choose Players: L'usuari podrà triar quin tipus de jugador vol que faci la partida.
- Choose K: En cas que l'usuari triï a dos jugadors "màquina" podrà triar la quantitat de partides que vol que faci. Després de cada partida, es tornaran a llistar els problemes perquè pugui canviar-lo, si així ho vol, per al següent partit.
- Play: L'usuari podrà jugar la partida o simular-la, en cas que ambdós jugadors siguin "màquines".
- Pause: En qualsevol moment de la partida, l'usuari podrà parar el joc.
- Reanudate: Amb la partida pausada, es podrà retornar a jugar en qualsevol moment.
- Save: Amb la partida pausada, l'usuari podrà guardar-la per tal de continuar més endavant.
- Quit: Amb la partida pausada, l'usuari podrà abandonar la partida, però es preguntarà si vol guardar la partida abans d'abandonar.

2.2 Problem Managment

- List Problem: L'usuari podrà veure en tot moment els problemes enregistrats al sistema.
- Load Problem: L'usuari podrà carregar un problema i veure el taulell que forma, per decidir si vol o no jugar-lo. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Clone Problem: L'usuari podrà clonar un problema per adaptar-lo a l'experiència que desitgi. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Modify Problem: L'usuari podrà modificar un problema ja existent al seu gust. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.

- Delete Problem: L'usuari podrà eliminar del sistema problemes que ja no vulgui. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Create Problema: L'usuari podrà crear un problema. Per fer-ho, podrà clonar o modificar un problema ja existent. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Validate Problem: Quan l'usuari hagi acabat de crear o modificar el problema el sistema haurà de verificar que té solució en els torns demanats. En cas que no ho sigui, no es guardarà al sistema i es notificarà a l'usuari.
- Save Problem: Si el problema crea és vàlid, llavors es guardarà a la base de problemes.

2.3 Ranking Managament

- List Ranking: L'usuari podrà veure el ranking de qualsevol problema. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Clear Ranking: L'usuari podrà buidar completament el ranking d'un problema. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.
- Delete Ranking: L'usuari podrà eliminar una puntuació del ranking d'un problema. Per fer-ho es llistaran tots els problemes disponibles al sistema perquè pugui escollir.