Open Chess Game Analyzer:

MEMORIA

Consultor

Miquel Angel Senar Rosell

Alumne

Josep M Sobrepere i Profitós

Introducció:

Aquest document detalla la feina que s'ha dut a terme per desenvolupar aquest projecte de Treball de Fi de Carrera, el projecte té com a títol "Open Chess Game Analyzer" i la finalitat del mateix és desenvolupar un aplicatiu lliure que sigui capaç d'analitzar partides d'escacs a partir d'un fitxer pgn.

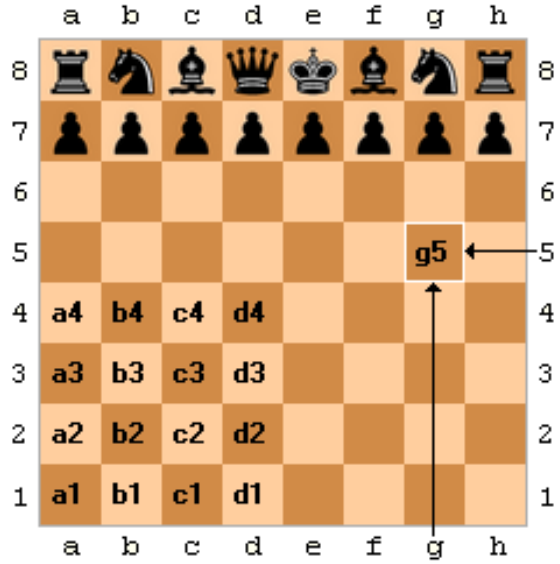
Es revisarà si s'han dut a terme els objectius que es van marcar en el pla de treball i s'explicarà les petites desviacions i imprevistos que no es van poder preveure a l'hora de fer el pla de treball.

Conceptes Previs:

- Notació algebràica (curta):

La notació algebraica és un sistema de notació en escacs que es fa servir per transcriure les jugades de manera que puguin ser reproduïdes amb tota precisió posteriorment.

Es basa en una estructura fixa, cada columna te una lletra que comença per la primera casella de l'esquerra del tauler (des del punt de vista de les blanques), que va de la a a la h, i cada fila es numera (també des del punt de vista de les blanques), començant per la fila 1 i fins a la fila 8; amb aquestes dades s'obtenen unes coordenades del tauler que permeten identificar cada casella de forma que no hi pugui haver error. La forma d'anotar la jugada és, en primer lloc, la inicial del nom de la peça que es mou, (tot i que quan es mou un peó no s'indica cap símbol per la peça, i s'entén que es un peó qui belluga), i després la coordenada de la casella on va, primer la columna i després la fila. Si es captura una peça, s'ha d'indicar amb el signe "x" entre la peça capturant i la peça capturada.



La notació algebraica més estesa és l'anglesa, la única diferència són les inicials que s'empren per identificar les peces, en aquesta versió les inicials que s'empren són:

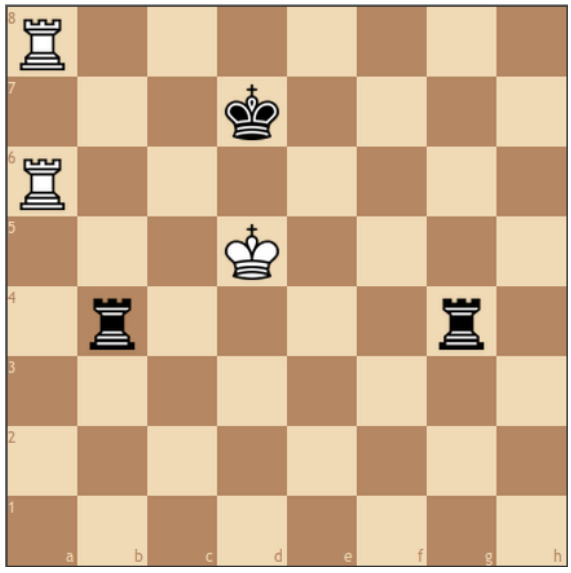
- R : per la torre (Rook)
- B: per l'alfil (Bishop)
- Q: per la dama (Queen)
- K: pel rei (King)
- N: pel cavall (Knight, s'empra l'N xq la "K" ja està assignada al rei)

En aquest document emprarem la versió anglesa de la notació algebraica, tot i que el projecte admetrà diferents notacions.

Si es fa escac es fa servir el signe (+) o escac i mat (#). Quan un peó corona, indiquem al final de la jugada un signe = i la peça resultant: "b8=Q" per una coronació a Dama. També pot ser que es coroni al mateix temps que es fa una captura, per exemple "bxc8=R" indicaria que el peó de la columna b captura la peça que hi ha a la casella c8 i corona una torre.

L'enroc curt se simbolitza 0-0, i l'enroc llarg 0-0-0.

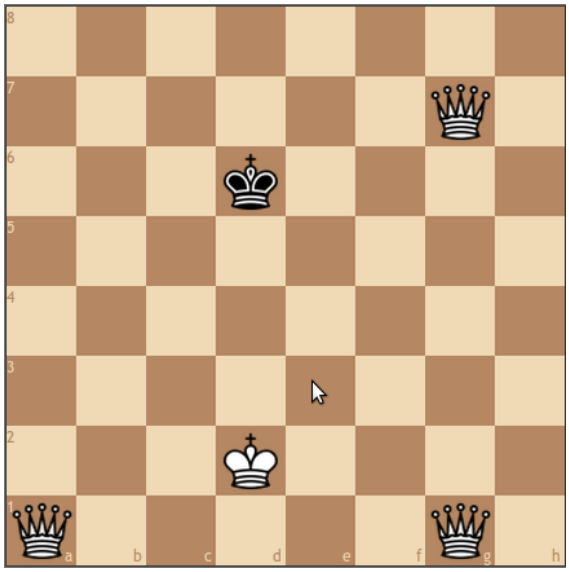
Per tant, en la notació algebraica curta se'ns indica a la casella de destí a la que s'està movent la peça, però no se'ns indica la casella de la que prové a no ser que sigui necessari. Per exemple: "Rd5" ens indica que s'està movent una torre a la casella d5. Però si hi ha més d'una peça del mateix tipus que pot accedir a aquella casella, se'ns proporcionarà a més la informació necessària per saber de quina peça es tracta. Normalment serà la fila o la columna que distingeix l'origen de les peces, i en rares ocasions seran ambdues. Per exemple, imaginem que tenim aquesta posició:



Si blanques volen moure la torre de la casella a6 a la casella a7 (que a més és mat), no poden fer-ho escrivint únicament Ra7, tal i com es faria normalment, hauran de fer-ho de la següent manera: R6a7 (de fet seria R6a7#). Si fos el torn de negres i les negres volguessin moure la torre que hi ha a la casella b4 a la casella d4, ho haurien de fer de la següent manera: Rbd4+.

D'altre banda un cas molt poc habitual, però que s'ha de tenir en consideració seria el següent: imaginem que blanques han coronat 2

dames (a més de la que ja tenien), i que arriben a una posició com aquesta, on li toca moure a blanques:



Si blanques volen moure una de les 3 dames a la casella d4, hauran d'indicar exactament de quina d'elles es tracta ja que n'hi ha 3 que tenen accés a aquella casella, per exemple ho haurien de fer de la següent manera: Qg7d4+.

- **Captura al pas:**

S'assumeix que el lector està familiaritzat amb el joc dels escacs, i que per tant coneix les normes del mateix. De tota manera, per tal de fer entendre alguns dels reptes del projecte s'explicarà la captura al pas de forma detallada, ja que acostuma a ser una norma desconeguda entre aquells que no juguen als escacs de forma habitual.

La captura al pas es tracta d'una captura especial que es pot fer només immediatament després que un jugador mogui un peó dues caselles endavant a partir de la seva posició inicial, moment en què un peó rival podria capturar-lo exactament igual com si s'hagués mogut només una casella en lloc de dues. En aquesta situació, la idea és que la captura és possible "mentre passa" per la primera casella, de manera que la posició resultant és la mateixa que si el peó s'hagués mogut només una casella i el peó rival l'hagués capturat normalment. Cal destacar que la captura al pas només és possible en el torn immediatament posterior al moviment inicial del peó rival; en cas de no fer la captura al pas en aquell moment, es perd el dret a fer-ho.

Vegem-ne un exemple:

1) Aquesta és la posició inicial, i li toca moure a negres:

2) Suposem que negres mouen d5, després de d5 el taulell queda així:

3) Observis que el peó de negres ha mogut dues caselles de cop, passant pel costat del peó de la columna e de blanques. Per tant en aquesta jugada, blanques tenen el dret de capturar amb el seu peó e el peó d de negres en la casella d6 de la següent manera:

La idea és que un peó no pot passar pel costat d'un altre peó del color contrari sense que aquest tingui el dret de capturar-lo.

- Desglòs de tasques del projecte:

L'objectiu ha estat desenvolupar un aplicatiu que a partir d'un fitxer PGN que contingui una o varies partides d'escacs, l'usuari estableixi les diferents opcions d'anàlisis, i el programa amb el suport d'un motor d'escacs UCI analitzarà la o les partides i retornarà un arxiu PGN amb els resultats de l'anàlisi. A més del motor UCI, i tot i que no es va especificar aquest objectiu inicialment, s'ha trobat adhient emprar també llibres d'opertures lliures (els que publica Toga2) que empren el protocol Polyglot. A continuació es detallen els diferents reptes que això comporta:

- **Lector de partides (línies principals) d'arxius PGN**: El primer repte ha estat desenvolupar un mòdul que sigui capaç d'extreure la informació que interessa de l'arxiu PGN l'usuari facilita. Els arxius PGN segueixen un protocol que permet inserir comentaris a les jugades, diferents variants a certes jugades i demés marques i senyals que s'han de descartar de cara a l'anàlisi. Per tant aquesta funcionalitat desenvolupada principalment a la classe PGNBasicReader.java, dins del package "util", el que fa és seleccionar únicament la informació de la partida que és rellevant per l'anàlisi, i recullir al mateix temps la informació de la partida que serà rellevant a l'hora de retornar l'anàlisi (data, hora, lloc de la partida, nom dels jugadors, etc). Per fer-ho s'ha combinat l'us d'expressions regulars, i pels nodes anhidats s'ha desenvolupat una funció pròpia.

- **Traductor bidireccional de Notació Algebraica a Notació Llarga**:

Aquesta ha estat, sens dubte, la funcionalitat més complexa de desenvolupar. Els arxius PGN estan en notació algebraica curta, però els motors d'escacs UCI no entenen aquesta notació, només entenen la notació algebràica llarga (coordenada d'inici, coordenada final). Per tant caldrà matenir un taulell que enregistri les posicions de la partida, i que entengui quines jugades són vàlides, quines no, si la posició es escac, si hi ha un peo al pas, les coronacions, si la posició és escac i mat, etc, per tal de poder traduïr d'una notació a l'altre. Més endavant s'explicarà amb detall la complexitat d'aquesta funcionalitat.

Aquesta funcionalitat ha estat desenvolupada al 100% i testejada molt a fons, les classes que implementen aquesta funcionalitat són bàsicament la classse MoveParser.java (del package utill) i la classe Board.java (del package board).

- Analitzador: Els motors d'escacs UCI són capaços d'analitzar qualsevol posició d'una partida d'escacs i retornar "el marcador" de la partida en aquell moment (en anglès l'score). El "marcador" el que fa és retornar l'avantatge que un bàndol té sobre l'altre en aquella posició, per exemple un marcador +1,5 indicaria que, segons el motor d'escacs, les blanques tenen una avantatge clara que podria venir donada, per exemple: per tenir un peó de més i una millor posició.

Tradicionalment a les peces d'escacs se'ls hi atribueix un valor estàtic, que vol dir que en una posició equilibrada aquell és el valor de cada peça, així doncs un peó val 1 punt, l'alfil i el cavall en valen 3, la torre en val 5 i la dama en val 9 (el rei en val infinit), ara bé: aquest valor és estàtic i les posicions fan que els valors es converteixin en dinàmics, per exemple en una posició tancada sense diagonals possiblement el cavall tindrà un valor més alt que l'alfil. Sigui com sigui, els motors d'escacs avaluen totes les variants de la posició i retornen una puntuació en centipeons, si el valor és negatiu vol dir que les negres tenen avantatge i si és positiu vol dir que les blanques tenen avantatge. Podria donar-se el cas (es dona molt sovint) que un bàndol tingui més peces sobre el taulell, però que l'altre bàndol estigui guanyant: ja sigui perquè té un atac molt fort que si l'altre bàndol vol evitar haurà de sacrificar peces de pes, o perquè s'està a punt de donar mat, o perquè té una combinació inevitable per guanyar més peces de les que li falten, els motius poden ser molt diversos. De qualsevol manera aquesta funcionalitat dels motors d'escacs permet comparar el marcador de cada jugada que s'ha dut a terme a la partida, amb el marcador que s'hagués obtingut fent la millor jugada que proposa el motor, si la diferència és escassa vol dir que la jugada que s'ha dut a terme és prou bona, però com més gran sigui la diferència més greu serà l'error. Això permetrà detectar les jugades s'hauran de senyalar com a dolentes, i proposar jugades alternatives. Per tant l'analitzador bàsicament serà un modul que vagi comparant (amb els paràmetres que se li donin) la puntuació de les jugades dutes a terme, amb la puntuació de les jugades que el motor d'escacs hagués jugat i detectar així els diferents tipus d'errors i inacuracitats que s'han dut a terme, així com les línies alternatives.

S'ha desenvolupat un alalitzador que empra el protocol UCI per comunicar-se amb el motor d'escacs que s'hagi seleccionat, i que extreu tota la informació necessària de cada posició per poder evaluar quines han estat les jugades en les que el motor d'escacs pot proposar alternatives interessants. A més, per fer que l'analitzador funcioni de manera més optima, s'ha desenvolupat una funcinalitat que mira si la posició és una posició conenguga, en terminologia escaquista "de llibre", per les jugades de l'apertura, d'aquesta

manera hi haurà menys jugades a analitzar pel motor d'escacs i això permetrà que es pugui destinar més temps a analitzar cada jugada.

- Interfície gràfica: S'ha desenvolupat una interfície gràfica senzilla que proporciona les funcionalitats necessàries per dur a terme l'anàlisis de la partida, a més l'aplicació utilitza l'estandar i18N i per tant és 100% multillengua.

Les funcionaliats que ofereix són: gestionar els diferents motors d'escacs (mostrar els ja instal·lats, afegir i treure), seleccionar l'arxiu PGN d'origen, l'arxiu PGN destí, el motor d'escacs que es vol emprar, establir les opcions d'anàlisis i que mostri una barra de progrés mentre l'anàlisi s'està duent a terme. Les opcions que disponibles per l'anàlissi són:

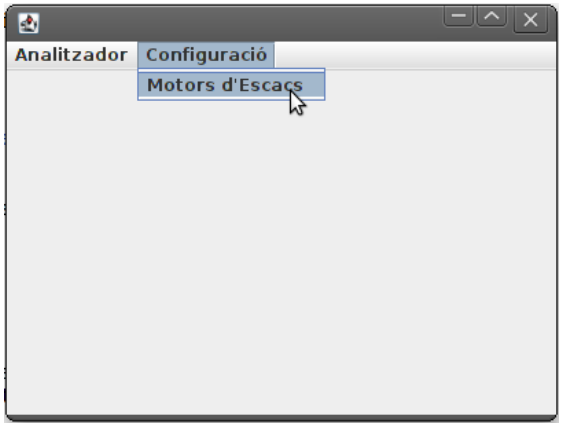
- Temps màxim d'anàlisis (en minuts): aquest paràmetre serà opcional, si es deixa en blanc s'analitzaran totes les jugades fins que el motor d'escacs cregui que ja no pot trobar una jugada millor.

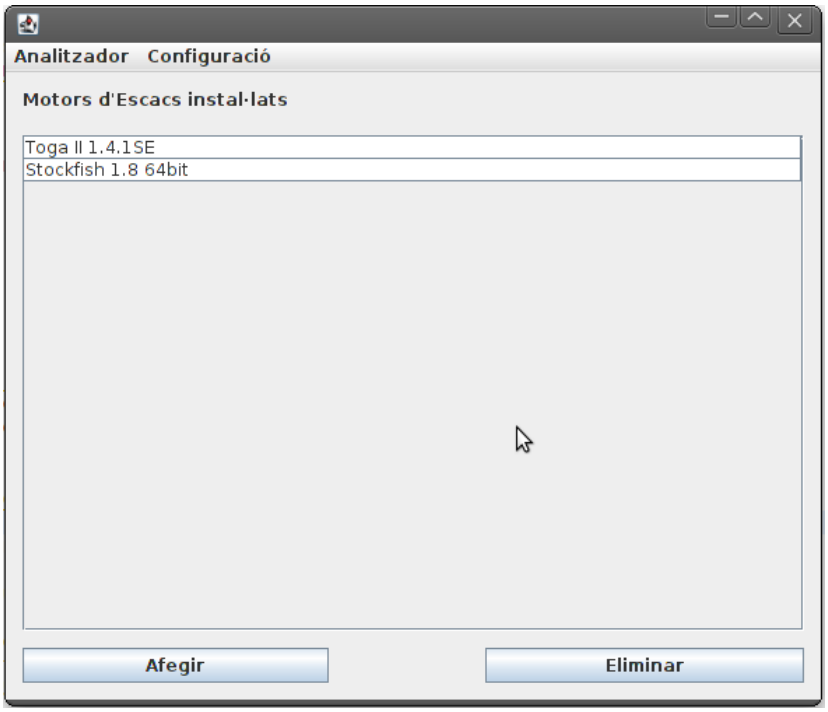
- Llindar d'error (en centipeons): tal i com ja s'ha explicat prèviament: és necessari determinar a partir de qui punt la diferència entre la puntuació òptima segons la jugada proposada pel motor d'escacs i la jugada duta a terme a la partida, és considera un error. En el seu defecte serà de 35 centipeons.

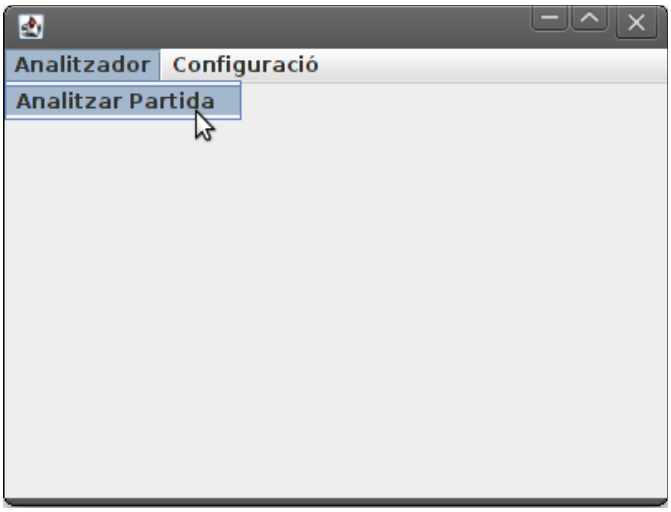
- Màxim de variacions a retornar: Es possible que l'usuari vulgui acotar el número màxim de variacions que vol que l'analitzador li retorni de la seva partida. Si estableix un número es prioritzaran les variacions de les jugades més dubtoses.

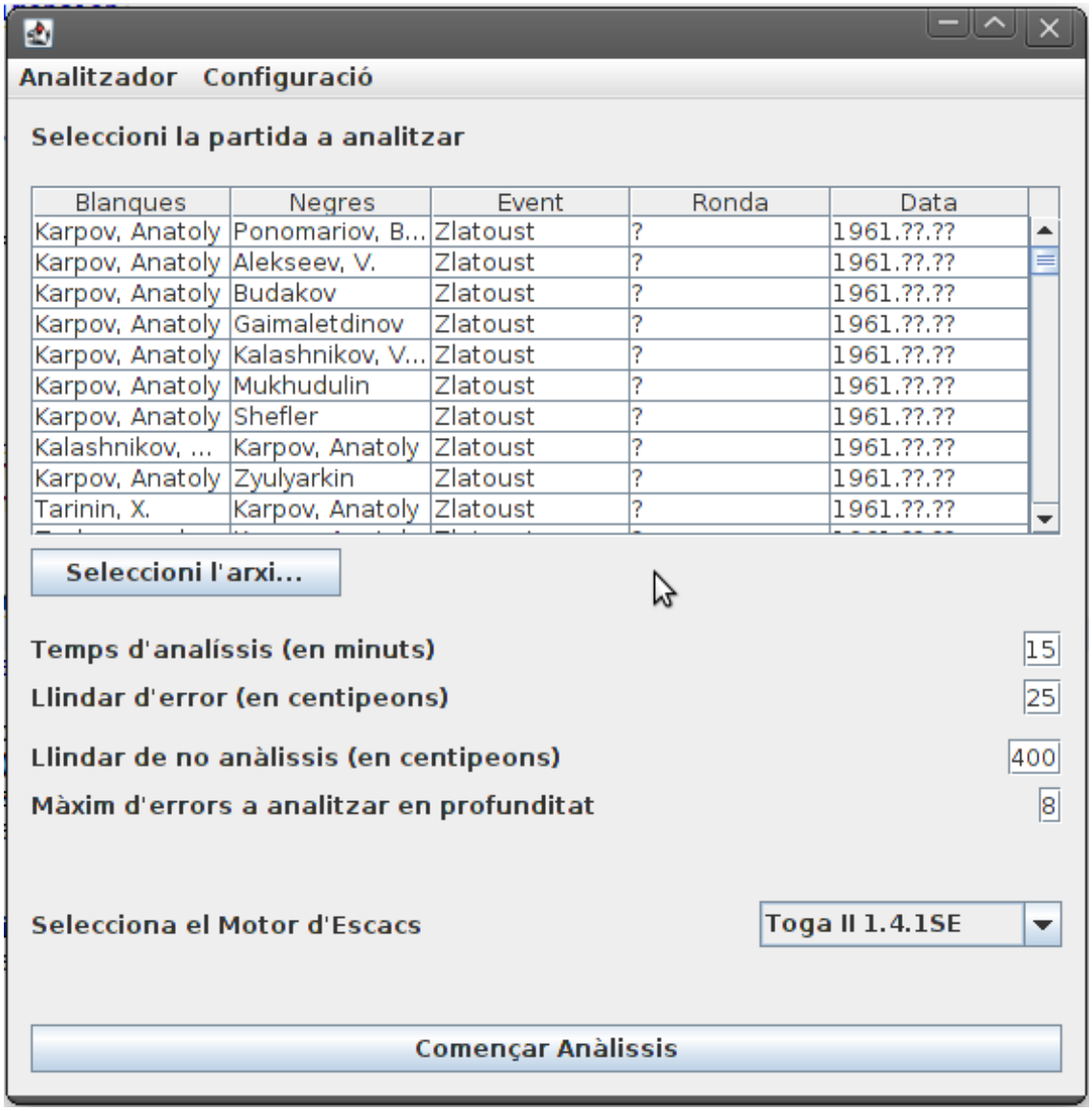
- Llindar de no anàlisi: Estableix a partir de quina puntuació en centipeons ja no cal continuar analitzant, normalment quan és té una avantatja significativa ja no cal continuar l'anàlisi, si un jugador és un màster i aconsegueix una avantatja d'1,5 punts potser ja no li caldrà continuar l'anàlisi perquè sap del cert que amb una avantatja com aquesta ja té la partida guanyada, però per a un jugador més feble potser aquest llindar haurà de ser de 4 o 5 punts, per tant es donarà l'opció a l'usuari de triar aquest llindar.

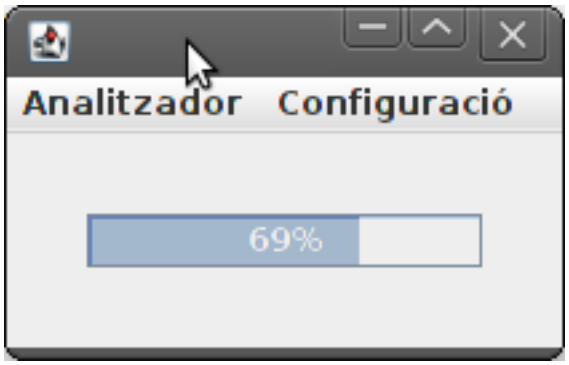
A continuació és mostren un seguit de pantalles que pretenen il·lustrar el funcionament d'aquest mòdul:


Analitzador
Configuració
Motors d'Escacs


Analitzador
Configuració
Motors d'Escacs instal·lats
Toga II 1.4.1SE
Stockfish 1.8 64bit
Afegir
Eliminar


Analitzador
Configuració
Analitzar Partida


Analitzador Configuració
Seleccioni la partida a analitzar
Blanques Negres Event Ronda Data
Karpov, Anatoly Ponomariov, B... Zlatoust ? 1961.??.??
Karpov, Anatoly Alekseev, V. Zlatoust ? 1961.??.??
Karpov, Anatoly Budakov Zlatoust ? 1961.??.??
Karpov, Anatoly Gaimaletdinov Zlatoust ? 1961.??.??
Karpov, Anatoly Kalashnikov, V... Zlatoust ? 1961.??.??
Karpov, Anatoly Mukhudulin Zlatoust ? 1961.??.??
Karpov, Anatoly Shefler Zlatoust ? 1961.??.??
Kalashnikov, ... Karpov, Anatoly Zlatoust ? 1961.??.??
Karpov, Anatoly Zyulyarkin Zlatoust ? 1961.??.??
Tarinin, X. Karpov, Anatoly Zlatoust ? 1961.??.??
Seleccioni l'arxi...
Temps d'anàlissis (en minuts) 15
Llindar d'error (en centipeons) 25
Llindar de no anàlissis (en centipeons) 400
Màxim d'errors a analitzar en profunditat 8
Selecciona el Motor d'Escacs Toga II 1.4.1SE
Començar Anàlissis


Analitzador Configuració
69%

- Descripció de les classes i packages del projecte:

- Package beans:

Dins del package "beans" hi trobem els JavaBeans necessaris per desenvolupar l'aplicatiu.

D'una banda tenim els enum:

- **Color.java**: Conté els possibles col·lors: Blanc o Negre.

- **ChessPiece.java**: Conté les diferents peces: peó, torre, cavall, alfil, dama i rei.

I d'altre banda tenim les classes:

- **Piece.java**: Combina els dos enum descrits anteriorment, representa de les peces del taulell.

- **Position.java**: Indica a partir de les coordenades una posició del taulell.

- **Move.java**: Aquest JavaBean és una mica més complexe, conté tota la informació relativa a una jugada/moviment. Concretament conté les següents propietats:
    - "Position from": indica la posició d'origen.
    - "Position to": indica la posició de destinació.
    - "Piece piece": indica la peça que s'ha mogut.
    - "Piece capturedPiece": en cas que s'hagi capturat una peça del bàndol rival.
    - "ChessPiece piecePromotion": S'hi s'ha coronat un peó, indica la peça en la que s'ha convertit.
    - "boolean check": si aquesta jugada ha generat un escac contra el rei contrari.
    - "boolean mate": si aquesta jugada ha generat és escac i mat.
    - "boolean takeEnPassant": si aquesta jugada és una captura al pas.

- "boolean pawnEnPassant": si aquesta jugada és un peó que pot ser capturat al pas pel bàndol contrari.

- "String shortAlgebraicNotation": text indicant la notació algebraica curta de la jugada.

- "int number": Indica el número de la jugada dins de la partida.

- "Move nextMove": per facilitar la navegabilitat els moviments estan enllaçats amb els moviments anteriors i posteriors.

- "Move previousMove": ídem.

- "AnalysisInformation analysis": En cas de que aquest moviment hagi estat analitzat, aquesta variable enllaça amb el bean que conté la informació de l'anàlisi.

– **AnalysisInformation.java**: Aquest JavaBean, directament associat a l'anterior, conté la informació que el motor d'escacs ens proporciona després d'analitzar la posició resultant després d'un moviment. La informació que conté és la següent:
  - "Move move": jugada que s'examina.

  - "int mate": en cas que el motor d'escacs hagi trobat una combinació que forci escac i mat, el número de jugades necessàries per forçar el mat.

  - "int score": la puntuació que el motor d'escacs li dona a la posició resultant després de la jugada.

  - "int differenceScoreBestMove": la diferència entre la puntuació de la posició resultant d'aquesta jugada i la puntuació de la millor jugada proposada pel motor d'escacs.

  - "List<Move> bestContinuation": a partir de la posició resultant, la combinació de millors jugadades proposades pel motor d'escacs.

– **Game.java**: Representa una partida, les propietats de la classe són:
  - "HashMap<String, String> info": Un hashmap amb la informació de la partida (Torneig, nom dels jugadors, etc)

- "LinkedList<Move> moves": Les jugades de la partida.
- "int score": la puntuació que el motor d'escacs li dona a la posició resultant després de la jugada.
- "Board board": un taulell amb la posició de la jugada que s'està examinant.
- "String rawGame": si la partida s'ha creat a partir d'un arxiu PGN, el contingut original de l'arxiu per aquella partida.

- Package board:

– **Board.java**: Una de les classes més crítiques del projecte. Aquesta classe representa el taulell d'escacs, i disposa de tota mena de funcions per manipular el taulell i per proporcionar informació sobre la posició del mateix. Les funcions més importants són:

- "void resisterMove(Move move)": rep una jugada per paràmetre i enregistra la jugada al taulell.
- "Position searchPositionPiece(ChessPiece kind, Position to, Color turn, String help)"

Tal i com ja s'ha explicat en la notació algebraica curta se'ns indica a la casella a la que s'està movent una peça, però no se'ns indica la casella de la que prové, però els motors d'escacs no entenen la notació algebraica curta, i se'ls hi ha de passar la coordenada d'origen i la coordenada de destí de cada moviment. Per tant ens caldrà una funció que a partir de la informació de la notació algebraica curta ens proporciona, ens retorni la posició d'origen, i això és exactament el que fa aquesta funció pública, amb el suport d'altres funcions privades. Els paràmetres que aquesta funció rep són: el tipus de peça que s'està movent, la coordenada a la que s'està movent, el color de la peça i el l'ajuda proporcionada per la notació algebraica curta, si és el cas (si hi ha dos peces del mateix tipus que tenen accés a la casella de destí, la notació algebraica proporciona informació addicional per triar la correcta).

Aquesta funcionalitat no és gens trivial degut a varies complicacions que es poden presentar, a mesura s'expliquin la resta de funcions és detallaran aquestes possibles complicacions.

- “Position[] getPiecesWithAccesToPosition(Position pos, ChessPiece kind, Color color, int xEnPassant,boolean checkPin)”

Aquesta és una de les funcions més importants. Aquesta funció es pot utilitzar per diferents propòsits, a grans trets el propòsit de la mateixa és obtenir les posicions de totes les peces d'un bàndol que tenen accés a una casella en concret, explicant els diferents paràmetres s'entendran les diferents maneres en les que aquesta funció pot ser emprada:

- Position pos: la casella en la que volem comprovar quines peces hi tenen accés.

- ChessPiece kind: aquest paràmetre pot ser null, si es passa null es cercaran totes les peces que tenen accés a la casella, del contrari si s'indica el tipus de peça, la recerca és limitarà a aquell tipus de peces.

- Color color: aquest paràmetre no pot ser null, indica el color de les peces amb accés a la casella.

- int xEnPassant: indica si la jugada anterior ha generat algun peó que pugui ser capturat al pas, si és així s'indicarà la columna en la que es troba aquest peó.

- boolean checkPin: indica si volem cercar peces que poden desplaçar-se en aquella casella (si el valor és cert), o si volem saber quines peces estan controlant aquella casella (si el valor és fals), en el segon cas també es retornaran les peces que estan clavades però que controlen una casella.

Per exemple, imaginem que el taulell està en una posició com aquesta:

I ens han passat la jugada Nd4, per trobar el cavall que s'ha mogut a la casella d4 emprarem aquesta funció, si no disposéssim d'aquest paràmetre la funció ens tornaria les posicions del cavall en b3 i del cavall en f3, però notis que en realitat el cavall en f3 no pot moure's a la casella d4 ja que està "clavat" ("pin" en anglés) degut a l'alfil negre que hi ha a la casella c6 (el cavall no es pot moure ja que seria una jugada il·legal: en escacs no pots posar el teu propi rei en posició d'escac ja que en escacs el rei mai pot ser capturat, la partida s'acaba en posició d'escac i mat, però mai s'arriba a capturar el rei).

Així doncs quan vulguem comprovar quines peces poden moure's a una casella concreta, no n'hi haurà prou en comprovar si hi tenen accés, també haurem de comprovar si estan clavades, notis que si blanques moguessin el cavall que hi ha a b3 a la casella d4, la notació algebraica per la jugada seria Nd4, però si l'alfil que hi ha a la casella c6 estigués a la casella b5, aleshores la notació per la jugada seria Nbd4.

D'altre banda hi ha situacions en les que volem saber quines peces estan controlant una casella, encara que no s'hi puguin moure, imaginem que al taulell hi tenim una posició com aquesta:

I el motor d'escacs ens proposa que la millor jugada és moure el cavall des de la casella e2 a la casella f4, el motor d'escacs ens proporcionaria la següent informació: e2f4, i ara haurem de convertir aquesta informació en notació algebraica curta, i per tant comprovar si és escac o escac i mat. En moure el cavall a la casella f4 el rei està en posició d'escac, ara, però, haurem de comprovar si és escac i mat, i

el primer pas per fer-ho serà mirar si el rei té caselles on desplaçar-se, en aquest cas haurem de comprovar quines peces controlen les caselles disponibles pel rei: les caselles g4 i h4 estan controlades pels peons, i la casella g5 està controlada pel cavall en f4, que tot i que no pot desplaçar-se a g5 perquè està clavat, controla la casella, per tant el rei no té caselles disponibles per escapar de l'escac, com que a més l'atac no pot ser bloquejat, ni la peça atacant capturada, és escac i mat. Notis que en aquest segon cas no volem saber quines peces poden desplaçar-se a una casella concreta, sinó quines peces controlen aquella casella.

- "Position[] pieceSearch(boolean recursive, Position position, ChessPiece kind, Color color, int yIncrement[], int xIncrement[], boolean checkPin)": Aquesta funció és una funció de suport a la funció anterior, serveix per cercar les peces que tenen accés a una casella concreta, serveix per tots els tipus de peces, excepte pels peon.

- "Position[] pawnSearch(Position position, Color color, int xEnPassant, boolean checkPin)": Igual que l'anterior però per peons. Calia una funció especial pels peons ja que aquests tenen vàries propietats que els fan diferents a qualsevol altre peça: només es mouen endavant, capturen en una direcció diferent a la que es mouen, no controlen caselles a les que es poden moure, poden capturar al pas, etc.

- "boolean isPieceUnderAbsolutePin(Position piecePosition, Position pieceDestination)": comprova si una peça està clavada.

- "int isCheck(Color attackingSide, int xEnPassant)": comprova si la darrera jugada és escac, o escac i mat. Retorna -1 si no és escac, 1 means si és escac i 2 si és mat.

- "Position findKing(Color color)": retorna la posició del rei.

- "boolean isMate(Position kingPosition,Color attackingSide,Position[] checkingPieces, int xEnPassant)": comprova si la darrera jugada és escac i mat, aquesta funció és bastant més complexa. El primer que es fa es comprovar si el rei té caselles on moure, si te caselles on desplaçar-se no és mat i no calen més comprovacions. Si no té caselles on desplaçar-se, el primer que és comprovarà és si és un escac doble (dos peces amenaçant al rei al mateix temps), si és així és escac i mat, ja que en una jugada no és poden neutralitzar 2 atacants, si no és escac doble es comprovarà si la peça que ataca al rei pot ser capturada, si pot ser-ho no és mat, en cas

de que no pugui ser capturada es comprovarà si alguna peça pot bloquejar l'atac, si es pot bloquejar no és mat, sinó es pot bloquejar és mat.

• "Position[] kingAvailableSquares(Position kingPosition)": funció de suport a l'anterior, comprova les caselles que el rei té disponibles per desplaçar-se.

• "boolean canBlockAttacker(Position kingPosition, Position attackerPosition)" : funció de suport a l'anterior, comprova si es pot bloquejar la peça atacant.

• "boolean checkMoveEnPassant(Position from, Position to)": Comprova si una jugada està generant un peó al pas.
• void printIt(boolean fromWhitePOV): funció útil pel testeig, imprimeix la posició del taulell.

- Package util:

Aquest package conté classes amb mètodes estàtics que seran d'utilitat de forma transversal per la resta de l'aplicatiu.

- **MoveParser.java**: Aquesta classe conté 2 mètodes estàtics que serveixen per parsejar jugades a partir de notació algebraica curta (la dels arxius PGN), o de notació algebraica llarga (la dels motors d'escacs).
    - static Move fromShortNotation(String move, Color turn, Board board, int numberMove)
    - static Move fromLongNotation(String moveStr, Board board, int numberMove)
- **PGNBasicReader.java**: Aquesta classe conté els mètodes necessaris per llegir arxius PGN.
- **PGNBasicWriter.java**: Aquesta classe conté els mètodes necessaris per escriure l'arxiu PGN resultant.
- **Util.java**: Calaix de sastre per funcions útils que poden ser utilitzades des de qualsevol part de l'aplicatiu.

- Package engine:

Aquest package conté les classes que interactuaran amb el motor d'escacs.

- **ChessEngine.java**: Aquesta classe permet interactuar amb el motor d'escacs i obtenir la informació que sigui necessària.

- **Analizer.java**: Aquesta classe analitza una partida en funció dels paràmetres descrits a la definició del projecte.

- Package Polyglot:

Aquest package conté les classes necessàries per interactuar amb un llibre d'opertures Polyglot, en realitat aquestes classes són una adaptació del codi font original que proporciona de forma lliure el protocol polyglot. El repte més gran va ser el fet que aquest codi treballa amb unsigned integers, i java no té cap mena de suport per aquest tipus de dades, vaig haver de traduïr les classes originals que estaven codificades en C++.

- Package GUI:

Aquest package conté totes les classes necessàries per implementar la interfície gràfica.

- Package i18N:

Aquest package conté totes les classes necessàries per fer que la interficie gràfica sigui multiidioma, així com els arxius amb "messages" que contenen els texts en diferents idiomes.

- Revisió dels objectius proposats:

Considero que s'han dut a terme tots els objectius que s'han proposat.

- **Generals:**

- He aplicat molts dels coneixements que vaig adquirir en les assignatures de: Programació Orientada a Objectes, Enginyeria del Programari, Estructura de la informació i Tècniques de desenvolupament del programari. De fet, l'arquitectura de la interfície gràfica i la implementació de l'estàndard i18N, són una herència directa de l'aplicatiu que vaig desenvolupar per l'assignatura Tècniques de Desenvolupament del programari.

- S'ha encetat un projecte GNU públic que podrà ser ampliat, que servir per analitzar partides d'escacs. Aquest projecte pot ser ampliat, se li pot afegir un visualitzador de partides pgn, i moltes altres funcionalitats que el faran més atractiu, però la base, que és el més complicat ja està feta.

- **Específics:**

- Aquest projecte m'ha servit per entendre i familiaritzar-me amb els protocols UCI,PGN, i l'especificació Polyglot. S'ha demostrat aquest domini amb l'implementació dels diferents mòduls.

- S'ha desenvolupat un mòdul independent capaç de parsejar la notació algebràica, aquest mòdul pot ser molt útil per la programació d'altres aplicacions d'escacs. És el primer mòdul de codi obert en java que parseja de forma senzilla la notació algebràica.

- S'ha desenvolupat un aplicatiu capaç d'analitzar partides d'escacs, és una versió que només fa l'anàlissis i el desa en un arxiu pgn. Hi ha moltes eines per visualitzar arxius pgn en un taulell d'escacs. Aquesta funcionalitat quedava fora de l'abast d'aquest projecte, i podria ser interessant afegir-la en un futur. Però l'objectiu marcat inicialment s'ha assolit amb escreix.