**Soberango 0.07.1 explained (well… trying to be explained):**

 -Uses 8x8 matrix as board.

 -The moves generator do not runs over board, runs over pieces.

This pieces has values that could change in case of pawns.

 -Each piece has numbered all its possible moves. For example the queen have 56 possible moves. Pawns, the insignificant pawns, have 18 possible moves!

 -To avoid make always the same move in the same situation, before each turn the number of each piece and the order of moves of each piece are randomly changed. This helps too to, just by chance, find better moves and not be stuck always in the same option cause the order of moves to be study are always the same.

-At this stage uses just material evaluation.

 -Uses a kind of MiniMax, “AlphaBeta” without Beta! And iterative deepening.

 -Just could play m/t/0 time controls.

Structure:

 The engine starts in Soberango

 XBoard0 ask for time for boot

 Booteo boots the engine.

SorteaPiezas choice by random the number to each of piece (same for black and whites) and for moves of each piece to do not search always in the same order.

 Xboard reads GUI messages and print Soberango results or comments to GUI.

 ProgramaPrincipal manages depth and timer and launch the moves generator.

 MainGenerador is the moves generator. Do not make a list of possible moves at each position, its moves depth first selecting pieces in the order determined by SorteaPiezas.

 Regla50Movidas; TerceraRepeticion and MaterialInsuficiente take care of 50 moves rules, 3rd repetition and insufficient material. 50 moves rules is quite obvious how it works. 3er repetition track uses random numbers for each move/each piece/each square that leads to different numbers for each position allow engine to compares them to know if one is repeated. Insuficient material uses a little “tablebase” to some special cases such as King and 2 knights vs king.

 At the end of each branch, mainGenerador launch Busqueda that mostly do the MiniMax and the saving of best move of depth d to be use it in depth d+1.

 Busqueda launch Evaluador that makes de evaluation of position.

 At the return from Busqueda, MainGenerador launch AlfaBeta that makes the AlphaBeta cut.

 After return from AlfaBeta, MainGenerador launch VuelveAtras that makes the “unmove”.

 When the tree search was ended, MainGenerador return and ProgramaPrincipal launch MovidaSoberangoaGUI that print the messages Xboard will send to the GUI and MovidaSoberangoaTablero that actualizes the internal board.

Important variables:

 Tablero (board): have the positions of each piece at every moment of search.

 PiensaSoberango: have the state of the engine: thinking, yet end thiking but still not sent its result, not thinking but yet sent the previous result.

 SPm1 and Scp: store the relation between “logical numbers of pieces and uses casue its random seting.

 Piezas: number of each of the 32 pieces.

 Valor: the value of each piece: 1 pawns; 2: knights; 3 bishops; 4; rooks; 5: queens; 6: kings kq; 7: kings k-; 8: kings –q; 9: kings --; 10: en passant pawns.

 Movimientos: the number of different moves each piece have. Not allowed, all. For example 8 for knights.

 Path: to track the Principal Variation

 Nodo: keeps the score for each branch

 ML: actual PV

profundidad: actual depth for search.

quienmueve: turn: 1 whites, -1 blacks.

Score: score.

ColorSoberango: wich pieces the engine use: 1 whites, -1 blacks.

tMejorMovida: piece that moves in best move from previous depth.

mMejorMovida: move that moves in best move from previous depth.