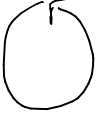


Chess++ → main()

↓
GenerateBoard() → returns board position
from FEN, or normal board

↓
Decide 2P/Engine game
i.e. white/black, start gameplay loop

(?)
Engine Move() →  Player Move() → returns board position
from player input
↓
Checkmate/Resign
ends game

ask player for move input
in algebraic notation, or occupied
square

↓
ParseInput() → returns a chess move
from player input

↓
MakeMove() → returns board position after
move is made

from player input, get info about
piece, starting/ending square

if given occupied
square

GetPossibleMoves() → returns list of moves
that piece can make

↓
CheckCheck() → remove any moves that put
yourself in check

↓
print list of legal moves,
ask for new player input

if given move in
algebraic notation

FindLegalMove() → returns the legal move that
satisfies input (or complains if illegal/
ambiguous)

↓
GetPossibleMoves() → returns list of moves
that piece can make

↓
CheckCheck() → remove any moves that put
yourself in check

↓
- ensure only 1 legal move
exists that fulfills conditions