```
ETF MODEL +
name of attribute
feature -- attributes
    board: BOARD
 moves board: BOARD -- a seperate chess board only for showing a piece's moves
 no of pieces: INTEGER
 game mode: INTEGER -- 1: GAME SETUP, 2: GAME PROGRESS: 3: GAME OVER
game mode msg: STRING
game result msg: STRING
display moves mode: BOOLEAN
error: STRING
feature { NONE } -- initialization
  make
     create board.make
     -- initialization other variables
feature -- model operations
  set game mode(j: INTEGER)
  set error(m: STRING)
  default update -- checks game mode when the player makes winning move
feature -- model commands start here
  setup_chess(p: STRING; row: INTEGER; col: INTEGER)
 valid game mode: game mode ≡ 1
 valid board slot: is valid slot(row, col)
 slot not already occupied: board.get(row, col) = "."
  start game
 require
 valid game mode: game mode ≡ 1
  moves(row: INTEGER; col: INTEGER)
 valid game mode: game mode \equiv 2 -- game has to be in progress for this method to be called
 valid board slot: is valid slot(row, col)
 slots occupied: ¬ is vacant slot (row, col)
  move and capture(r1: INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER)
 require
 valid game mode: game mode \equiv 2
  reset_game
 require
 valid game mode: game mode = 2 or game mode = 3
feature -- queries
   ret game mode msg(j: INTEGER): STRING
  is valid slot(r: INTEGER; c: INTEGER): BOOLEAN
  is_vacant_slot(r: INTEGER; c: INTEGER): BOOLEAN
  is move possible(r1: INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER): BOOLEAN
  is block exists(r1: INTEGER; c1: INTEGER; r2: INTEGER; c2: INTEGER): BOOLEAN
  out : STRING -- print method
```

HISTORY_LIST[COMMAND

