CSE 259 - Logic in Computer Science

Recitation-6

Project 2: Chess - Part 1

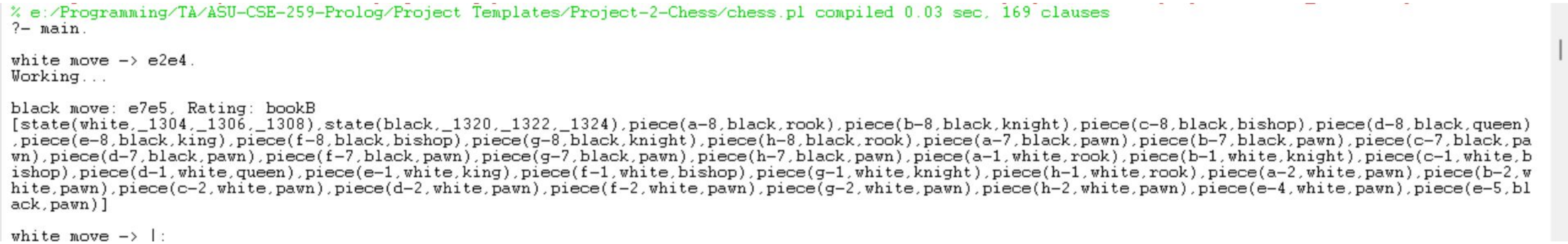
Waqar Hassan Khan

Project 2

- Implement a Chess program
- 3 Tasks
    1. Visualize the chess board
    2. Write codes for playerA so that it can move on its own. PlayerB codes are already there!
    3. Use PlayerA's code to play against PlayerB

Project 2

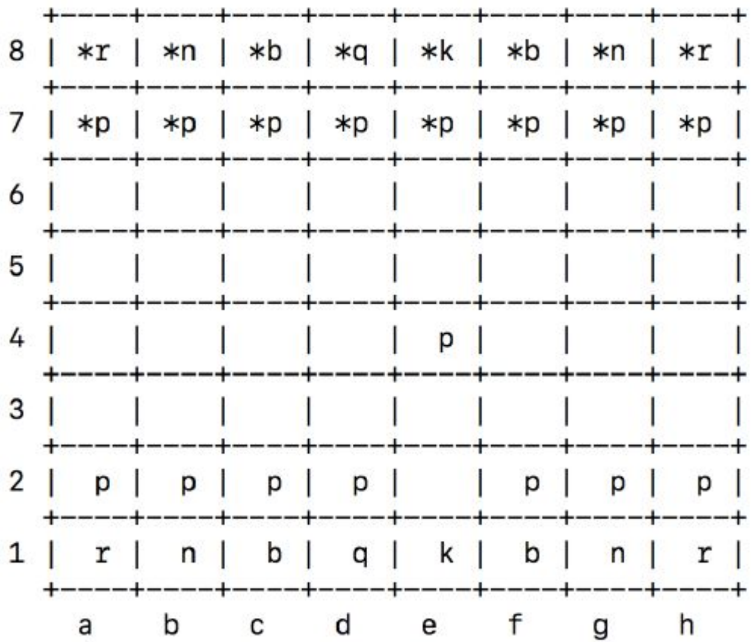
- We will call **main.** from the console
- If the template is ran, the following output is seen: It asks for whites move and the black moves on it's own



```
% e:/Programming/TA/ASU-CSE-259-Prolog/Project Templates/Project-2-Chess/chess.pl compiled 0.03 sec, 169 clauses
?- main.

white move -> e2e4.
Working...

black move: e7e5, Rating: bookB
[state(white,_1304,_1306,_1308),state(black,_1320,_1322,_1324),piece(a-8,black,rook),piece(b-8,black,knight),piece(c-8,black,bishop),piece(d-8,black,queen)
,piece(e-8,black,king),piece(f-8,black,bishop),piece(g-8,black,knight),piece(h-8,black,rook),piece(a-7,black,pawn),piece(b-7,black,pawn),piece(c-7,black,pa
wn),piece(d-7,black,pawn),piece(f-7,black,pawn),piece(g-7,black,pawn),piece(h-7,black,pawn),piece(a-1,white,rook),piece(b-1,white,knight),piece(c-1,white,b
ishop),piece(d-1,white,queen),piece(e-1,white,king),piece(f-1,white,bishop),piece(g-1,white,knight),piece(h-1,white,rook),piece(a-2,white,pawn),piece(b-2,w
hite,pawn),piece(c-2,white,pawn),piece(d-2,white,pawn),piece(f-2,white,pawn),piece(g-2,white,pawn),piece(h-2,white,pawn),piece(e-4,white,pawn),piece(e-5,bl
ack,pawn)]

white move -> |:
```

Project 2 - task 1

- Write codes so that the chase board is drawn visually



```
  +-----+-----+-----+-----+-----+-----+-----+-----+
8 | *r  | *n  | *b  | *q  | *k  | *b  | *n  | *r  |
  +-----+-----+-----+-----+-----+-----+-----+-----+
7 | *p  | *p  | *p  | *p  | *p  | *p  | *p  | *p  |
  +-----+-----+-----+-----+-----+-----+-----+-----+
6 |     |     |     |     |     |     |     |     |
  +-----+-----+-----+-----+-----+-----+-----+-----+
5 |     |     |     |     |     |     |     |     |
  +-----+-----+-----+-----+-----+-----+-----+-----+
4 |     |     |     |     |  p  |     |     |     |
  +-----+-----+-----+-----+-----+-----+-----+-----+
3 |     |     |     |     |     |     |     |     |
  +-----+-----+-----+-----+-----+-----+-----+-----+
2 |  p  |  p  |  p  |  p  |     |  p  |  p  |  p  |
  +-----+-----+-----+-----+-----+-----+-----+-----+
1 |  r  |  n  |  b  |  q  |  k  |  b  |  n  |  r  |
  +-----+-----+-----+-----+-----+-----+-----+-----+
     a     b     c     d     e     f     g     h
```

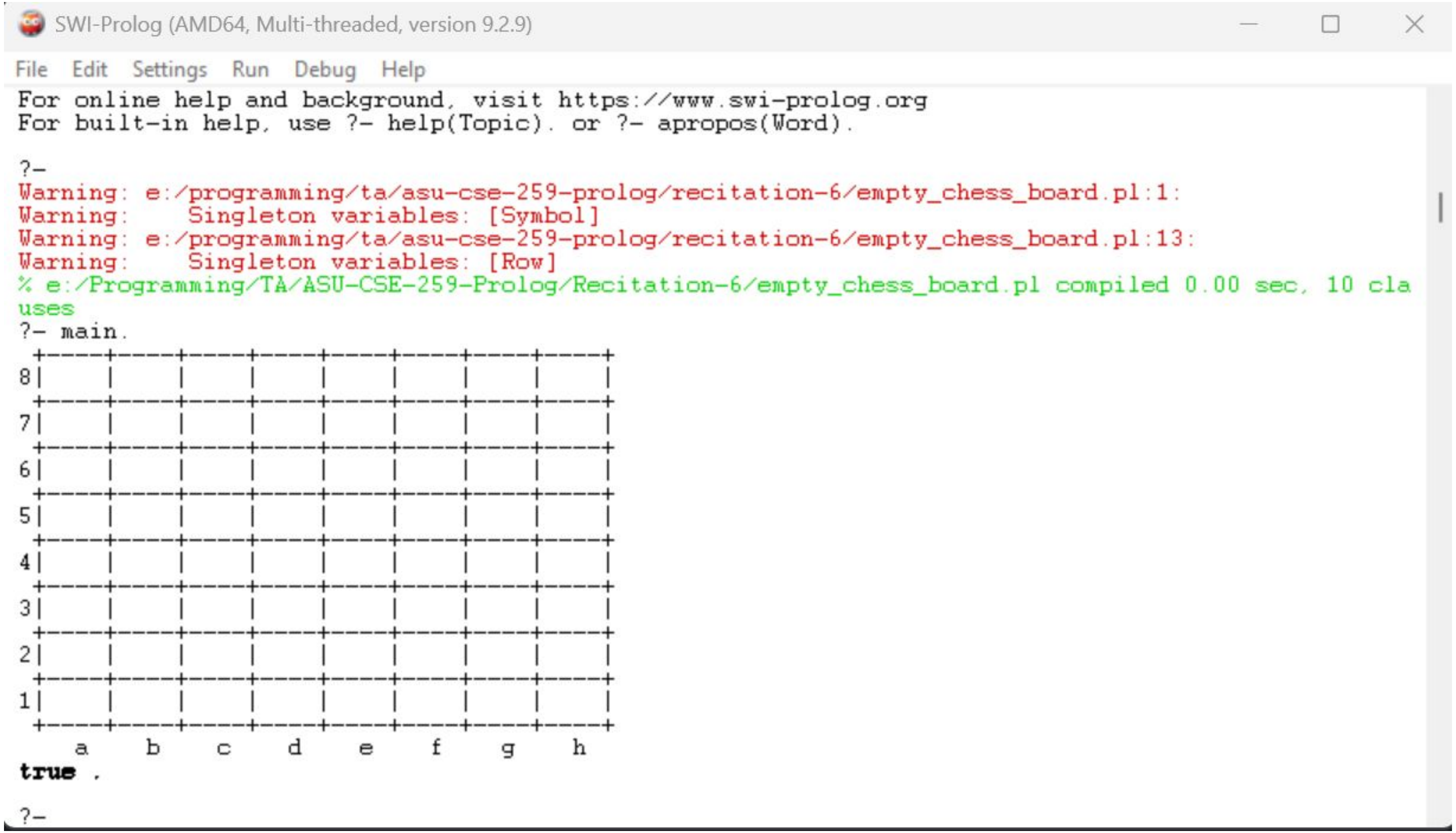
Project 2 - task 2

- Implement playerA’s code. Mimic the code for playerB.

Project 2 - task 3

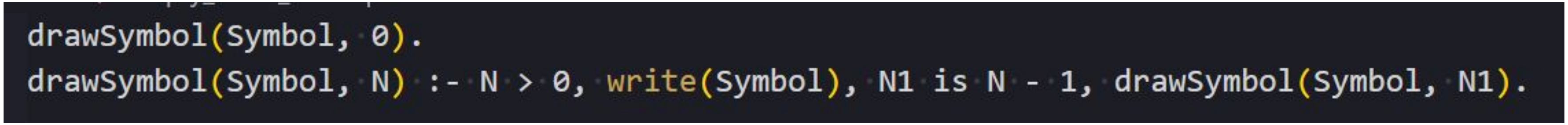
- Use playerA’s code to play against playerB. no need to write much code here. The challenge is to understand the main process of the chess program

Project 2 - Drawing the board



```
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: e:/programming/ta/asu-cse-259-prolog/recitation-6/empty_chess_board.pl:1:
Warning:    Singleton variables: [Symbol]
Warning: e:/programming/ta/asu-cse-259-prolog/recitation-6/empty_chess_board.pl:13:
Warning:    Singleton variables: [Row]
% e:/Programming/TA/ASU-CSE-259-Prolog/Recitation-6/empty_chess_board.pl compiled 0.00 sec, 10 cla
uses
?- main.
 +----+----+----+----+----+----+----+----+
8|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
7|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
6|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
5|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
4|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
3|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
2|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
1|    |    |    |    |    |    |    |    |
 +----+----+----+----+----+----+----+----+
    a    b    c    d    e    f    g    h
true .

?-
```

Project 2 - Drawing the board

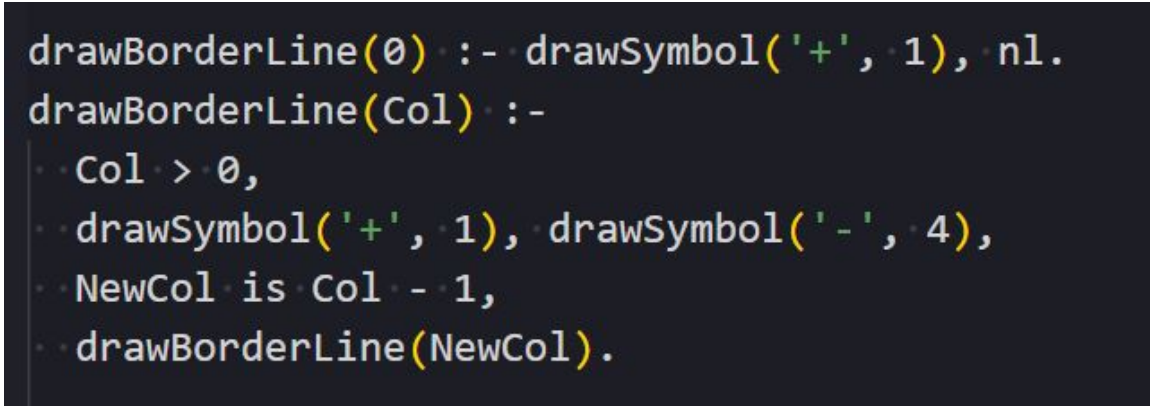


```
drawSymbol(Symbol, 0).
drawSymbol(Symbol, N) :- N > 0, write(Symbol), N1 is N - 1, drawSymbol(Symbol, N1).
```

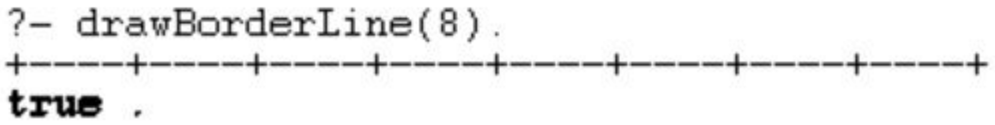
- Draws a characters for N times

Project 2 - Drawing the board

- Draws the borders of the chess board



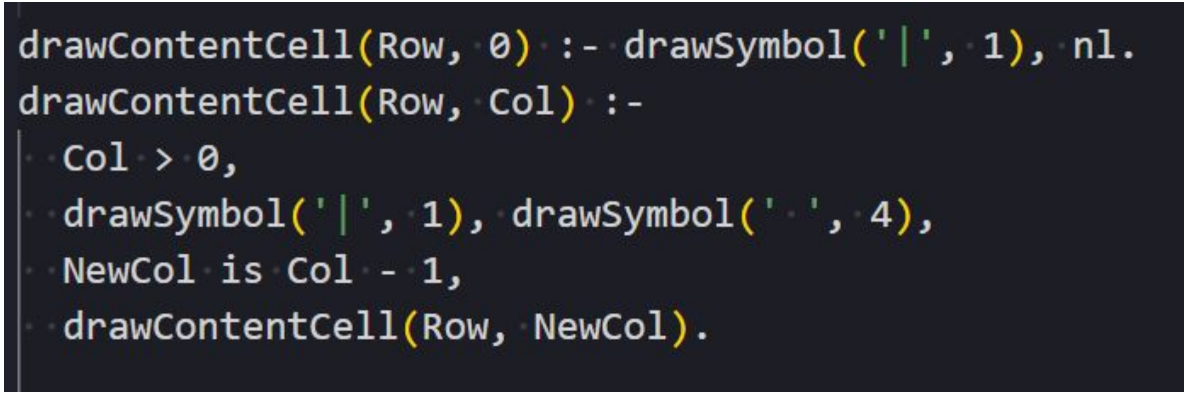
```prolog
drawBorderLine(0) :- drawSymbol('+', 1), nl.
drawBorderLine(Col) :-
    Col > 0,
    drawSymbol('+', 1), drawSymbol('-', 4),
    NewCol is Col - 1,
    drawBorderLine(NewCol).
```



```
?- drawBorderLine(8).
+----+----+----+----+----+----+----+----+
true .
```

Project 2 - Drawing the board

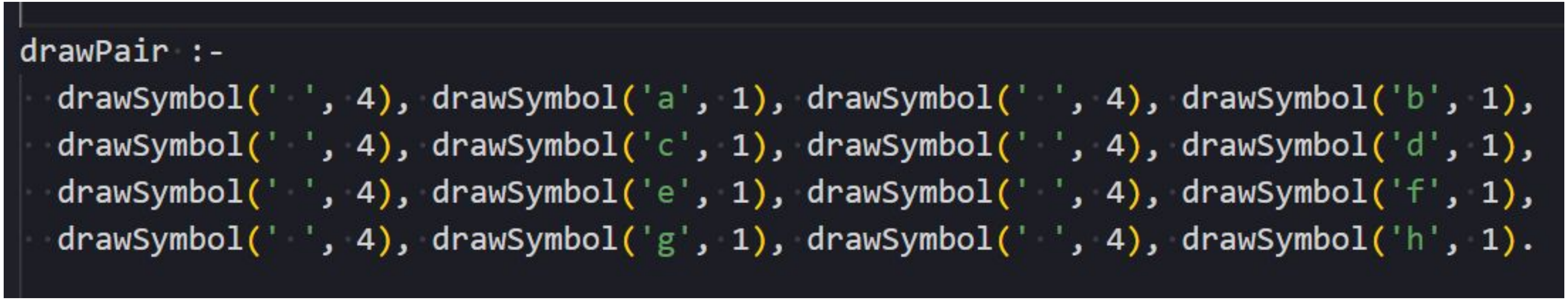
- Draws the cells where we will have the chess pieces



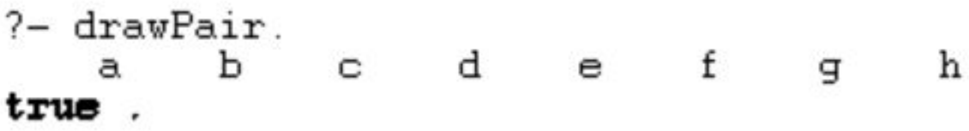
```
drawContentCell(Row, 0) :- drawSymbol('|', 1), nl.
drawContentCell(Row, Col) :-
    Col > 0,
    drawSymbol('|', 1), drawSymbol(' ', 4),
    NewCol is Col - 1,
    drawContentCell(Row, NewCol).
```

Project 2 - Drawing the board

- Does the numbering of cells



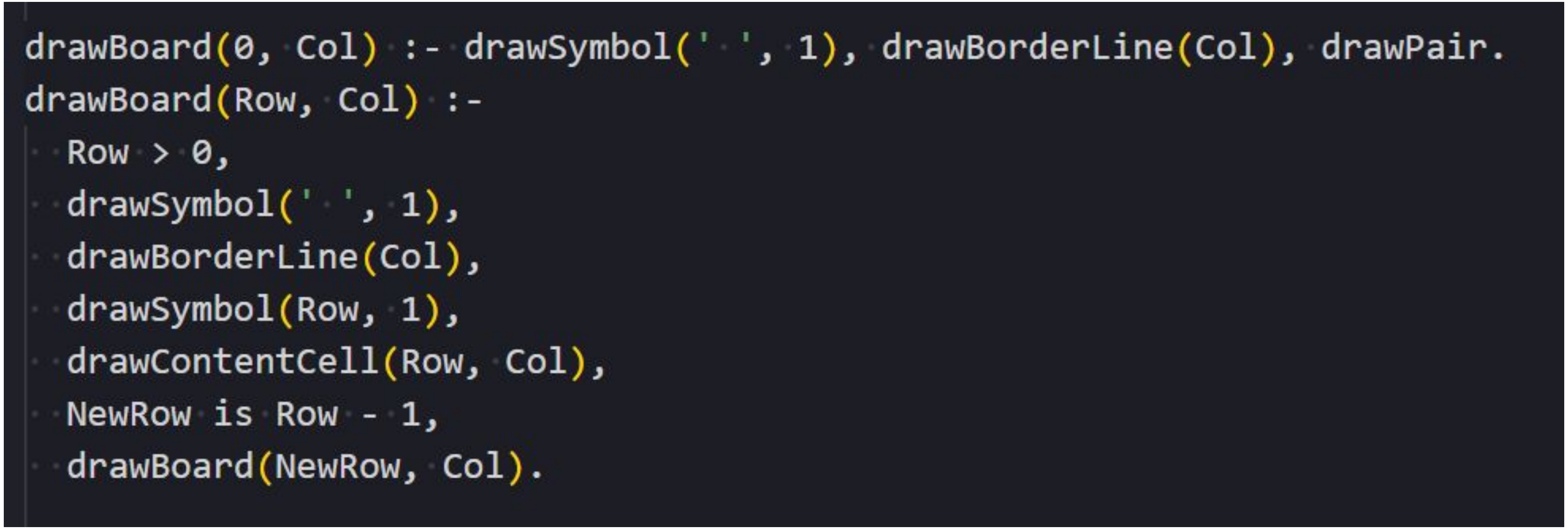
```
drawPair :-
  drawSymbol(' ', 4), drawSymbol('a', 1), drawSymbol(' ', 4), drawSymbol('b', 1),
  drawSymbol(' ', 4), drawSymbol('c', 1), drawSymbol(' ', 4), drawSymbol('d', 1),
  drawSymbol(' ', 4), drawSymbol('e', 1), drawSymbol(' ', 4), drawSymbol('f', 1),
  drawSymbol(' ', 4), drawSymbol('g', 1), drawSymbol(' ', 4), drawSymbol('h', 1).
```



```
?- drawPair.
    a    b    c    d    e    f    g    h
true .
```

Project 2 - Drawing the board

- The rule to draw the board



```
drawBoard(0, Col) :- drawSymbol(' ', 1), drawBorderLine(Col), drawPair.
drawBoard(Row, Col) :-
  Row > 0,
  drawSymbol(' ', 1),
  drawBorderLine(Col),
  drawSymbol(Row, 1),
  drawContentCell(Row, Col),
  NewRow is Row - 1,
  drawBoard(NewRow, Col).
```