```
% eight_queens_gui.pl
% SWI-Prolog version with GUI display using XPCE library.

:- use_module(library(pce)).
:- use_module(library(clpfd)).  % Needed for constraint logic

% Entry point
start :-
   queens(8, Qs),
   draw_board(Qs).

% Generate solution for N queens
queens(N, Qs) :-
   length(Qs, N),
   Qs ins 1..N,
   safe_queens(Qs),
   labeling([], Qs).

safe_queens([]).
safe_queens([Q|Qs]) :-
   safe_queens(Qs, Q, 1),
   safe_queens(Qs).

safe_queens([], _, _).
safe_queens([Q|Qs], Q0, D) :-
   Q0 #\= Q,
   abs(Q0 - Q) #\= D,
   D1 #= D + 1,
   safe_queens(Qs, Q0, D1).

% Draw GUI board
draw_board(Qs) :-
   length(Qs, N),
   new(Window, picture('8 Queens Problem')),
   send(Window, size, size(400, 400)),
   SquareSize is 400 / N,
   draw_squares(Window, N, SquareSize),
   draw_queens(Window, Qs, 1, SquareSize),
   send(Window, open).

% Draw chessboard squares
draw_squares(_, 0, _).
draw_squares(Window, N, Size) :-
   N > 0,
   N1 is N - 1,
   draw_row(Window, N1, N, Size),
   draw_squares(Window, N1, Size).
```

```
draw_row(_, _, 0, _).
draw_row(Window, Row, Col, Size) :-
   Col > 0,
   Col1 is Col - 1,
   X is Col1 * Size,
   Y is Row * Size,
   ( (Row + Col) mod 2 =:= 0 -> Color = white ; Color = black ),
   new(Box, box(Size, Size)),
   send(Box, fill_pattern, colour(Color)),
   send(Window, display, Box, point(X, Y)),
   draw_row(Window, Row, Col1, Size).

% Draw queens (with proper index tracking)
draw_queens(_, [], _, _).
draw_queens(Window, [Q|Qs], Index, Size) :-
   X is (Index - 1) * Size + Size/4,
   Y is (Q - 1) * Size + Size/4,
   new(Qimg, circle(Size/2)),
   send(Qimg, fill_pattern, colour(red)),
   send(Window, display, Qimg, point(X, Y)),
   NextIndex is Index + 1,
   draw_queens(Window, Qs, NextIndex, Size).
```

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- working directory(_, 'D:/Prolog').
tru
?-
tru
?-
tru
```