Shakti

Topic and group

For the first Practical Assignment (TP1) of Functional and Logic Programming 2023/2024 @ FEUP, we implemented the board game **Shakti** in Prolog. Our group is Shakti_7 from Class 13, composed by:

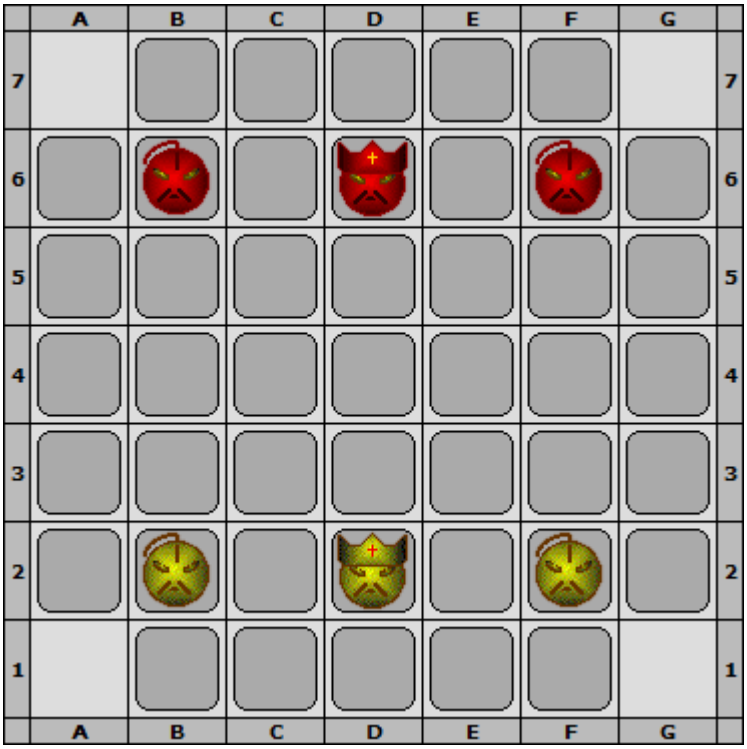
- João Mendes Silva Belchior - up202108777@up.pt (50%)
- José Francisco Reis Pedreiras Neves Veiga - up202108753@up.pt (50%)

Installation and execution

We need SICStus Prolog 4.8 to play this game. There, we should consult the shakti.pl file found in the src folder and type **play** to start running the game's code.

Description of the game

Shakti is a Chess variant invented by Christian Freeling in 1982. It is played on a 7x7 board. Initially each cell of the board is covered with a tile. Both players have 1 King and 2 Warriors. The initial position of the pieces is shown on the following picture:



The objective of Shakti is to checkmate the opponent's King (i.e. to attack the opponent's King in such a way that the opponent cannot escape from the attack and cannot capture the attacking piece). Stalemate is bypassed in Shakti (meaning, a player who puts his opponent in a "stalemate" position will immediately get another move). A draw is possible by 3-fold or mutual agreement.

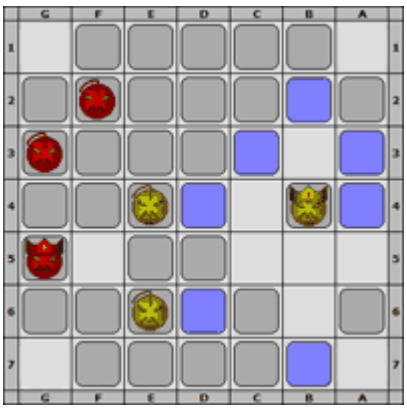
Players alternate moves, starting with the player controlling the white pieces. A player must move on his turn, unless he has no legal moves, in which case he passes his turn to the opponent. A piece can only move on tiled squares (during a game some tiles are removed as the result of particular moves).

A King is said to be "in check" if there is an opponent's Warrior on the same horizontal, vertical or diagonal as the King and there are no tiles between them (even if the attacking Warrior cannot move for some reason). A player may never leave his King "in check" at the end of his move. Check can be eliminated in one of the following ways:

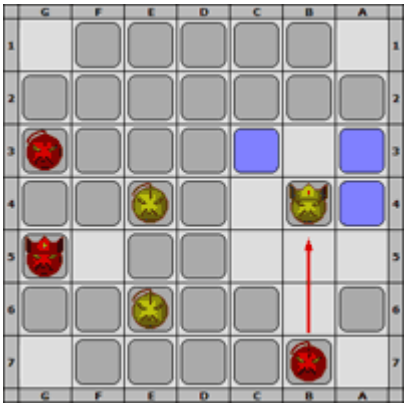
- The King can be moved to a cell that is not under attack.
- The attacking opponent's Warrior can be captured by the player's King (if doing so does not put the King in check).

It is forbidden to cause a "mutual check" situation when both Kings are on the same horizontal, vertical or diagonal and there are no tiles between them. All possible moves for each type of the pieces are explained below:

- A King, if not in check, may move to the first unoccupied tile in any of eight directions:

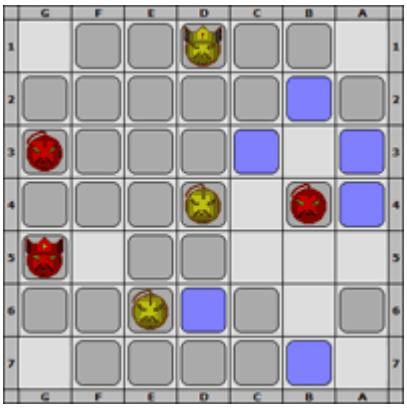


- If in check, a King is restricted to adjacent tiles. In this case the target tile may be either empty or occupied by an opponent's Warrior (if by capturing the attacking piece the King does not move in check):

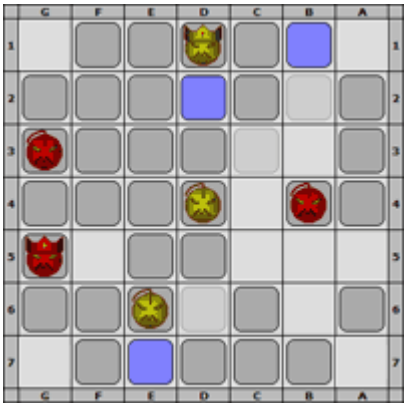


A Warrior can make two kinds of moves:

- A Warrior may move to the first unoccupied tile in any of eight directions:



- If both tiles are vacant a Warrior may jump over the first tile in any of eight directions and land on the second tile in the same direction. In this case the first tile is removed from the board. The removal of the first tile is compulsory. So if making such a move puts the player's King in check then the move is illegal:



Game Logic

Internal Game State Representation

Our game state is represented by the board and color of the player who has to move. The board is a list of lists and its atoms are empty, tile, white/black warrior and white/black king.

Game State Visualization

The program first displays the main menu, where the user can different game modes : Player vs Player, Player vs Computer or Computer vs Computer. Then the game loop starts and the initial board is displayed, asking the player with the white pieces to input its move.

Move Validation and Execution

With each move input, our program verifies if the coordinates of the move are in the bounds of the board. After that, it verifies if the player wants to move his own piece and not his opponent's, and accordingly with the rules of the game, such as moving a specific number of tiles and not leaving the king in check.

List of Valid Moves

We used the findall predicate to find all the valid moves in a given game state.

End of Game

Not implemented.

Game State Evaluation

Not implemented.

Conclusions

Implementing a game in Prolog was very different from what we did in other UCs using other programming languages.

Bibliography

- Rules of the game
- Rules pt. 2
- Play Shakti online