Rules:

1. Each player starts out with 12 playing pieces. One player has light colored pieces (called “white”) and the other has dark pieces (called “black” although the pieces may be red). Each player starts with their pieces laid out on the 12 dark squares nearest him or her
2. The player with the black pieces moves first. Randomly determine who gets the black pieces first. Use a coin toss (many authors say to “cast lots”). When playing a series of games, the players alternate who gets the black pieces.
3. “All jumping moves are compulsory.” Every opportunity to jump must be taken. In the case where there are different jump sequences available, the player may choose which sequence to make, whether it results in the most pieces being taken or not.
4. When a player’s piece lands in one of the squares at the far end of the board, its move ends there and it becomes a king.
5. A king can move or jump in any of the four diagonal directions within the limits of the board. A king cannot jump over one of the player’s own pieces.
6. A player wins by either capturing all the other player’s pieces or putting them into a position where they cannot move. A player can also win if the other player resigns or forfeits the game because of a violation of the rules.

Source: <http://www.darkfish.com/checkers/rules.html>

UML:

|  |
| --- |
| Gameboard |
| +Board: Struct  +spaces: SpacePtr  +LinkedList: struct  +head: NodePtr  +Player: char  +numPieces: int |
| drawBoard(LinkedListPtr p1, LinkedListPtr p2): BoardPtr  displayBoard(BoardPtr board, Print \*top, Print \*bottom): void  refresh(BoardPtr board, LinkedListPtr p1, LinkedListPtr p2, Print \*top, Print \*bottom): void  checkMove(BoardPtr, LinkedListPtr, LinkedListPtr, int, int, int, bool ): int  isOut(int, int): int  addPiece(LinkedListPtr, BoardPtr, int, int, bool): LinkedListPtr  removePiece(LinkedListPtr, BoardPtr, int, int, bool): LinkedListPtr  updateBoard(BoardPtr, LinkedListPtr, LinkedListPtr): void  takeTurn(BoardPtr, char\*, LinkedListPtr, LinkedListPtr, LinkedListPtr, LinkedListPtr, LinkedListPtr, Print, Print): int |

