Fuzzy-Logic Medieval Chess©

A Fuzzy-Logic Chess© Variation of Medieval Warfare

F-L Medieval Chess builds on F-L Chess to create game-play resembling a medieval battle, usingthe standard chess board. Pieces have different battle capabilities like their historical counterparts: mounted and armored knights and royalty, pikemen, infantry, and archers. The F-L Medieval Chess Capture Table shows how these different pieces fare in battle against each other. Moving and capturing are separate actions, and UP TO three actions may be taken in each turn).

CAPTURING PIECES Fuzzy-Logic Chess (F-L Chess) introduces uncertainty to the act of capturing a piece so that players must use *reasoning with uncertainty* (*fuzzy logic: probability*) in planning their strategies. The attacking player rolls a die to determine if a capture is successful. The die roll needed to capture a piece depends on the combination of the attacking piece and the defending piece, as shown in the Capture Table.

- When an attack is successful, the attacking piece moves into the square of the captured piece.
- When an attack is unsuccessful, the attacking piece remains in the square it attacked from.
- All pieces may attempt to capture an opposing piece in any direction (left, right, up, down, diagonal).

F-L Medieval Chess Capture Table
Die Roll Needed to Capture the Defending Piece

Attacking Piece	Defending Piece					
	King (1stguard)	Queen (2ndguard)	Knight (knights)	Bishop (pikemen	Rook (archers)	Pawn (infantry)
King (1stguard)	6,5,4	6,5,4	6,5,4	6,5,4	6,5	automatic
Queen (2ndguard)	6,5,4	6,5,4	6,5,4	6,5,4	6,5	6,5,4,3,2
Knight (knights)	6	6	6,5,4	6,5,4	6,5	6,5,4,3,2
Bishop (pikemen)	6,5	6,5	6,5	6,5,4	6,5	6,5,4,3
Rook (archers)	6,5,4	6,5,4	6,5	6,5	6	6,5
Pawn (infantry)	6	6	6	6,5	6	6,5,4

Mounted Knights and Royalty

In F-L Medieval Chess, the King, Queen, and Knights may each move in any direction, and do not have to move in a straight line. Not counting the starting square, but counting the final square, the **King and Queen may move three squares,** and the **Knights five squares**. They ***may not*** jump over or pass through an occupied square. The King and Queen represent the elite royalty, more heavily armored than knights, but slower.

Knights (only) may combine movement with a capture in the same action, attacking any adjacent enemy piece after their movement, **but subtract one from the die roll.**

The Infantry (Pawns and Bishop)

Pawns represent simple infantry with minimal training and arms, while Bishops represent pikemen with long pikes or halberds with formal training and experience.

- Pawns and Bishops may move and attack in a forward direction only, either directly ahead or to either forward diagonal toward the opposing player.
- Pawns are never promoted after reaching the eighth rank (the opponent's home row).
- Pawns **may not** move two squares forward when they have not yet moved in the game

Archers (Rook/Castle)

The Rook represents a company of archers, who may move a single square in any direction. Archers may attack any piece by shooting over up to two squares (not counting the square with the Rook or the square with the enemy piece).

Description From D2L Discussion

Project 4C: AI and Chess Variant

Available on Tuesday, January 7, 2020 12:48 PM EST

Develop a chess playing application, with a GUI that displays the board and allows the human player to move according to the rules of the game. You do not need to be a chess player to work on this project, chess familiarity is not required.

A chess variant will be the goal (attached file) which adds some new features and capabilities to standard chess rules.

Sponsor/Contact: Dr. Hoganson

You may write your application in Java or any other language/platform that produces a sharable game executable.

Clearly there are multiple aspects to this project: understanding the variant; developing a GUI to play the game; developing an AI system to play against a human opponent.

For the AI, you will use a distributed AI model! The "armies" of chess pieces for the AI is divided into three "corp":

- The "left" side bishop commands the three left pawns and the left knight.
- The "right" side bishop commands the three right pawns and the right bishop
- The king commands the queen, two rooks, and the remaining two center pawns (the king may delegate any of its pieces to be commanded by either bishop, at any time, based on your AI decision process.

Pieces scan their local areas for opportunities or threats, and provide the results of their observations to their commander (bishop or king). The commander (bishop or king) makes decisions based on the input from the the pieces it commands, and may also do its own scan of the board, and makes decision on the actions of its pieces (move or

attack, the knight and queen make both move and attack). The bishops may move themselves and engage in combat.

Start small, get your communication between pieces working. Then as time allows add more sophisticated planning ideas over two or more turns, and add identifying threats and possible responses. The AI system will need to scan the board looking for situations where the tactical rules will come into play.

For instance: scan the board to look for situations where the computer's more powerful piece can attack a less powerful enemy piece - where there is an advantage in the probability of a successful attack (the fuzzy logic).

Another example idea: scan the board to look for situations where the computer's pieces may be vulnerable to an attack by the human component, and look for a piece to move to adjacent to the vulnerable piece, in order to make a subsequent attack on the human player's piece.

Extra credit if you share a common turn play communication protocol, to allow two AIs to play against each other.