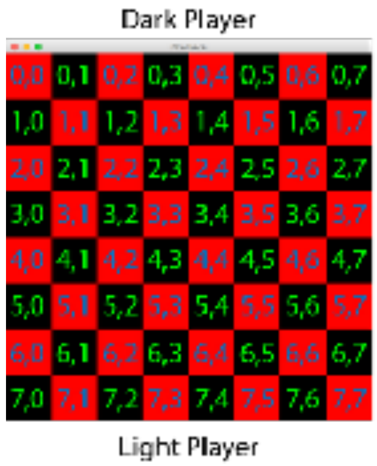
Checkers Data Model

Joe Wong
jlwxz8

Layout:

Since the Dark Player goes first we always orient the map with 0,0 on the dark player's side.



Legality of moves for dark player:

- A move is legal if the row number is exactly +1 from the current piece's row(CR) AND +/- 1 from the current piece's column (CC).
- No row values of less than 0 or greater than NUM_ROWS - 1 can be legal
- No col values of less than 0 or greater than NUM_COLS - 1 can be legal

Legality of Jumps for dark player:

- A light color piece must exist within the legal movement of current piece
- Row of jump space is CR + 2
- Col of jump space is CC + 2*(Captured piece's col - CC)
- Jump space must be empty

Legality of moves for light player:

- A move is legal if the row number is exactly -1 from the current piece's row(CR) AND +/- 1 from the current piece's column (CC).
- No row values of less than 0 or greater than NUM_ROWS - 1 can be legal
- No col values of less than 0 or greater than NUM_COLS - 1 can be legal

Legality of Jumps for light player:

- A dark color piece must exist within the legal movement of current piece
- Row of jump space is CR - 2
- Col of jump space is CC + 2*(Captured piece's col - CC)
- Jump space must be empty

Kings:

- A dark piece becomes a king if it reaches NUM_ROWS - 1.
- A light piece becomes a king if it reaches row 0.
- A king assumes all legal moves for a regular dark piece.
- A king assumes all legal moves for a regular light piece.

General Game Play:

- Dark player must always go first
- If a jump exists, current player must make a jump and any subsequent jumps
- A player loses when there are no legal moves or jumps
- A draw occurs when drawCounter is reduced to 0 from 50
  - drawCounter reduces by 1 for each turn in which no KING or JUMP occurs and is reset on King or JUMP event

Data Model FIELDS list:

- currentPlayer:Player
- darkPlayer:Player
- lightPlayer:Player
- jumpList[ ]:Jump
- moveList[ ]:Move
- darkChips[ ]: Chip
- lightChips[ ]: Chip
- boardState[ ][ ] - holds Chip(king, color) where color can be light, dark
- drawCounter:int
- currentPosition:Position(int, int)
- turnCounter:int

Data Model METHODS list:

- setCurrentPlayer(Player)
- getCurrentPlayer():Player
- setDarkPlayer(Player)
- setLightPlayer(Player)
- makeJumpList()
- getJumpList():Jump[ ]
- isJumpLegal():boolean
- makeMoveList()
- getMoveList():Move[ ]
- isMoveLegal():boolean
- setBoardState(Position, Chip)
- getBoardState(): Chip[ ][ ]
- setDrawCounter(int)
- getDrawCounter():int
- setCurrentPosition(Position)
- getCurrentPosition():Position
- setTurnCounter(int)
- getTurnCounter():int