Checkers Game Data Model Concept

By Nijaporn Hotrabhavananda

Checker Game Class:

Board : CheckerBoard

Player1 : Player

Player2 : Player

moveHistory : Map<Player, List<int[2][2]>>

selectedPieceColor : Map<Player,String>

Methods:

selectPieceType(Player) : void

startTheGame() : void

announceWinner() : Player

In the announceWinner() method, it will list all the legal moves and illegal move and also determine who will win, lose or draw.

Player Class:

playerName : String

numOfWins : int

numOfLoss : int

numOfDraws : int

matchesPlayed : List<String>

Methods:

getPlayerName()

setPlayerName()

getNumOfWins()

setNumOfWins()

getNumOfLoss()

setNumOfLoss()

getNumOfDraws()

setNumOfDraws()

Player class will keep track of numbers of wining, losing, and draws and machesPlayed.

Board class:

rectangle : Box[8][8]

removedPieces : List<Piece>

Methods:

movePiece(pieceColor, sourceX, sourceY, destinationX, destination) : boolean

checkForChecker() : boolean

removePiece(sourceX, sourceY)

This Board class will keep track of how the board get used.

Piece class:

color: String

Methods:

display()

tracePaths(sourceX, sourceY, destinationX, destinationY): PathTrace

This will keep track of all trace paths for pices.

King class extends Piece:

kingPiece: Piece

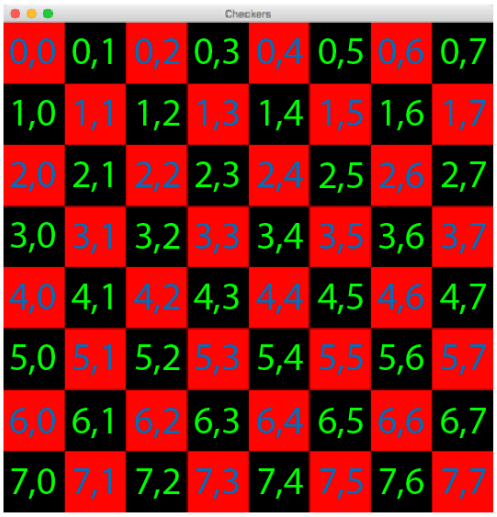
Methods:

tracePaths(sourceX, sourceY, destinationX, destinationY): PathTrace // Add the ability to move backward

This King Piece will extend out the ability from Piece class with backward.

- We will initialize all pieces by the Board class, but the Board and the pieces doesn't know about players.

Dark Player



Light Player