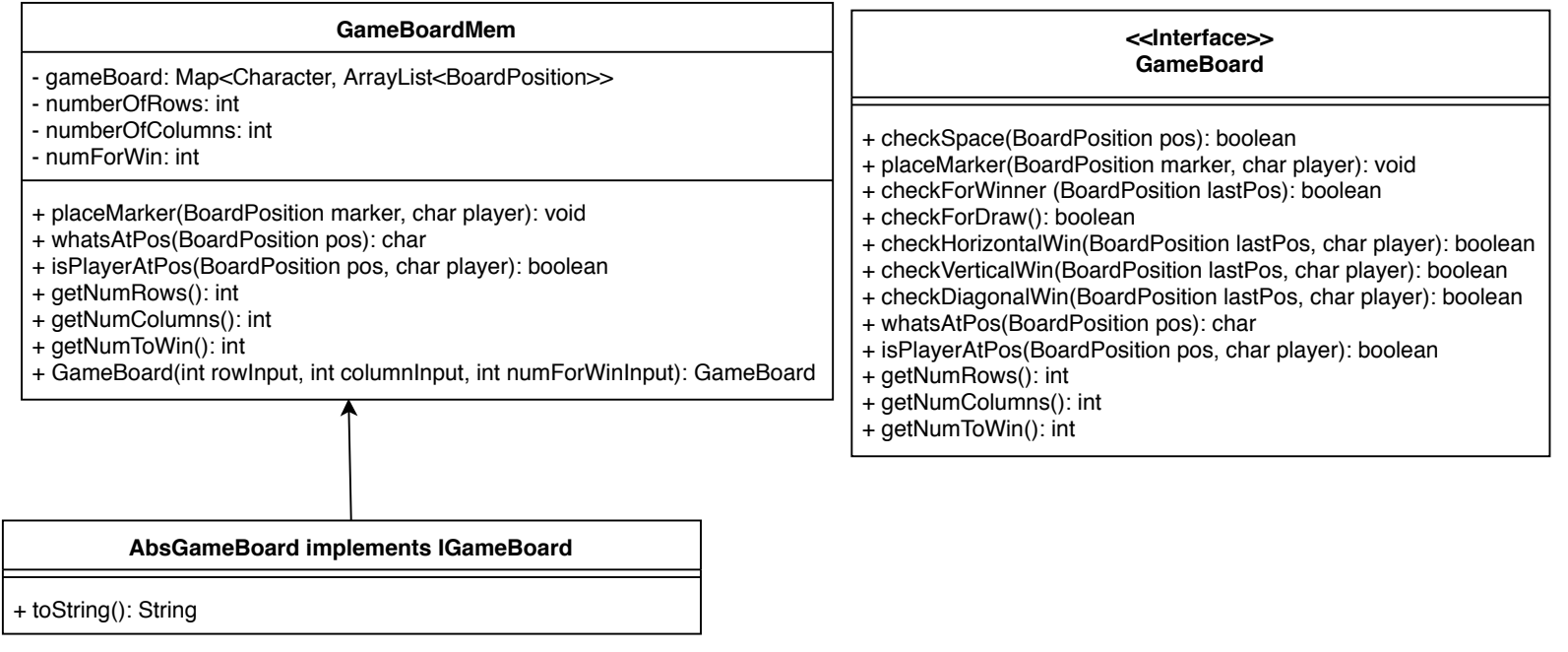
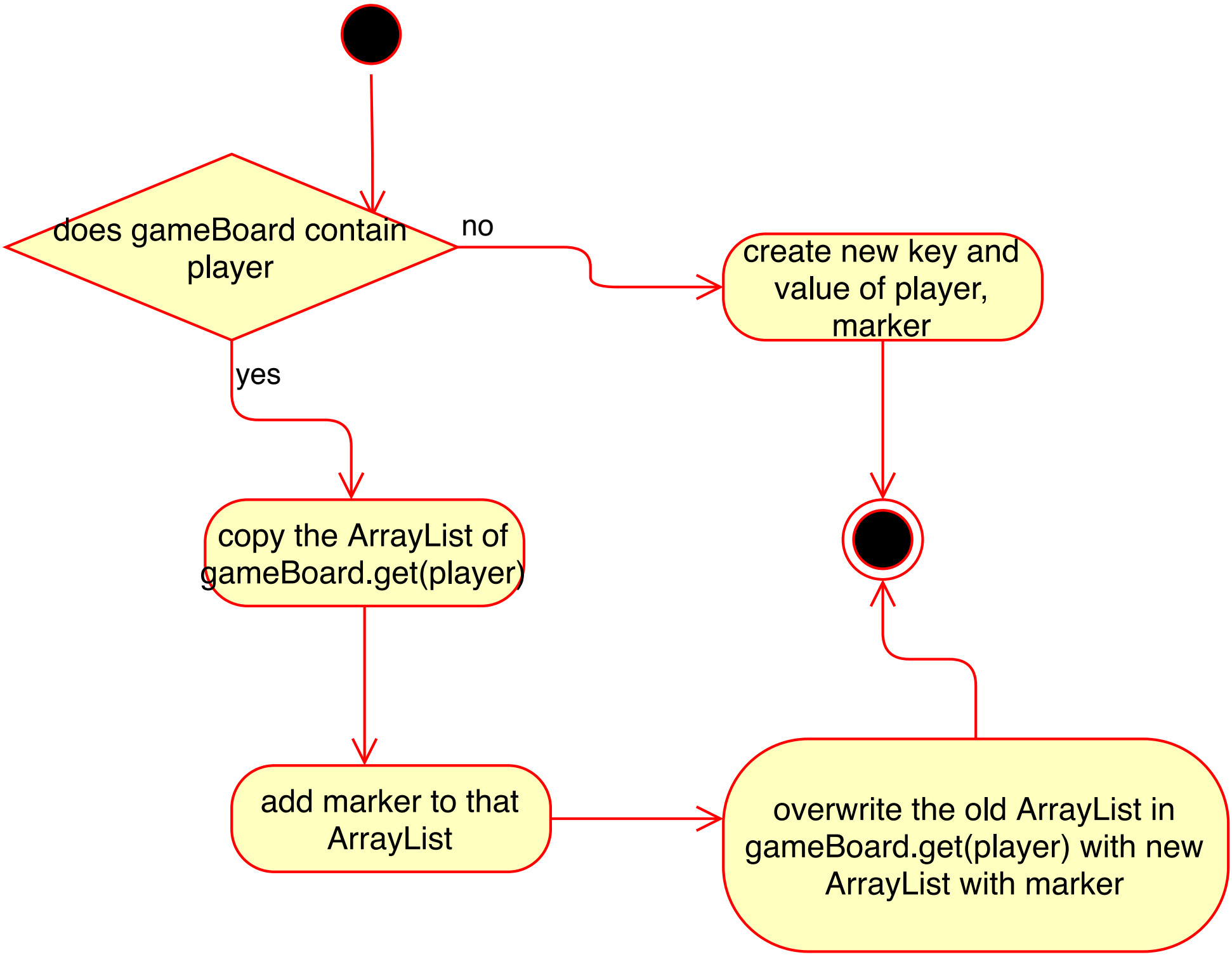
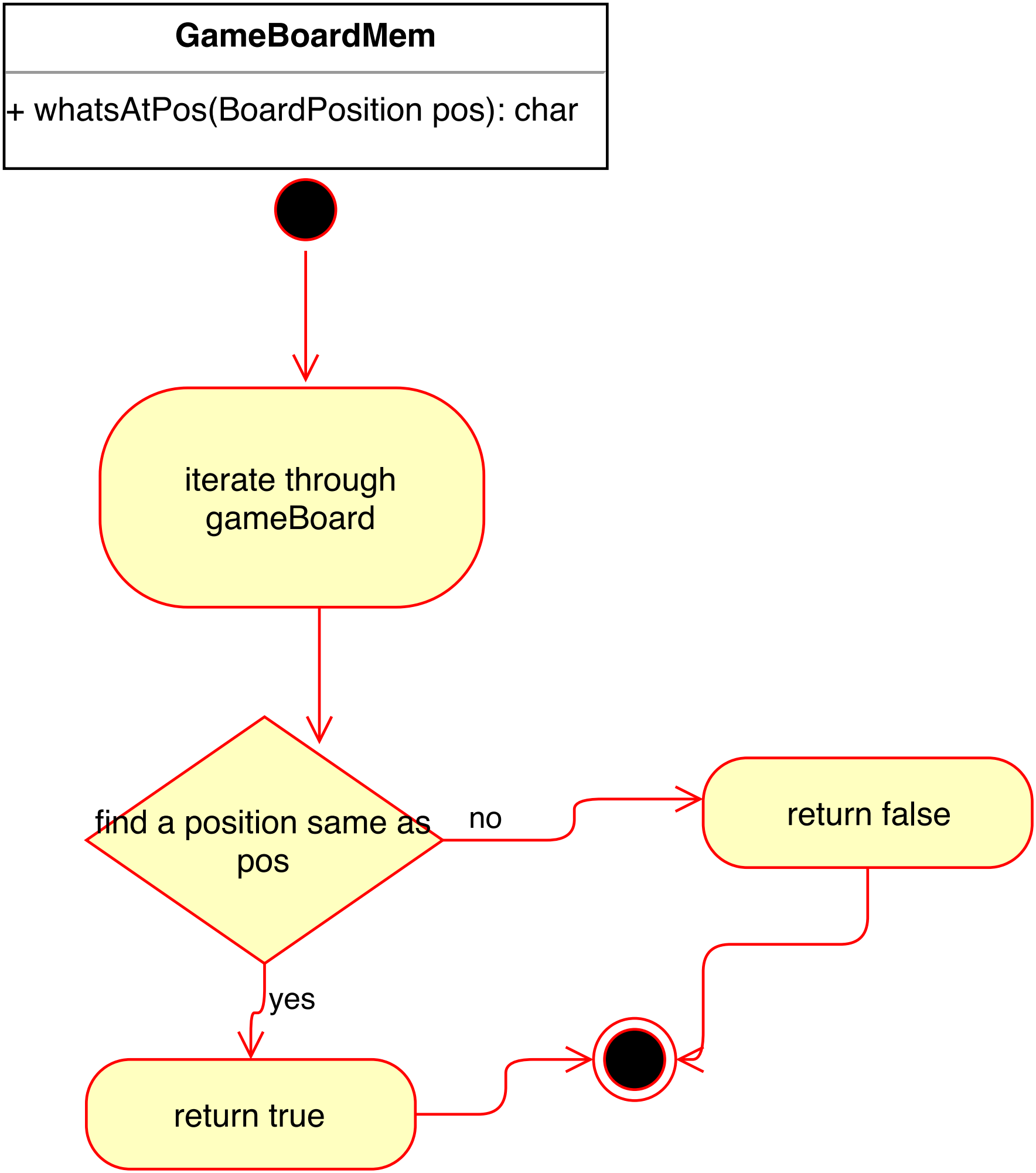

GameBoardMem
- gameBoard: Map<Character, ArrayList<BoardPosition>>
- numberOfRows: int
- numberOfColumns: int
- numForWin: int
+ placeMarker(BoardPosition marker, char player): void
+ whatsAtPos(BoardPosition pos): char
+ isPlayerAtPos(BoardPosition pos, char player): boolean
+ getNumRows(): int
+ getNumColumns(): int
+ getNumToWin(): int
+ GameBoard(int rowInput, int columnInput, int numForWinInput): GameBoard
<<Interface>>
GameBoard
+ checkSpace(BoardPosition pos): boolean
+ placeMarker(BoardPosition marker, char player): void
+ checkForWinner (BoardPosition lastPos): boolean
+ checkForDraw(): boolean
+ checkHorizontalWin(BoardPosition lastPos, char player): boolean
+ checkVerticalWin(BoardPosition lastPos, char player): boolean
+ checkDiagonalWin(BoardPosition lastPos, char player): boolean
+ whatsAtPos(BoardPosition pos): char
+ isPlayerAtPos(BoardPosition pos, char player): boolean
+ getNumRows(): int
+ getNumColumns(): int
+ getNumToWin(): int
AbsGameBoard implements IGameBoard
+ toString(): String

GameBoardMem

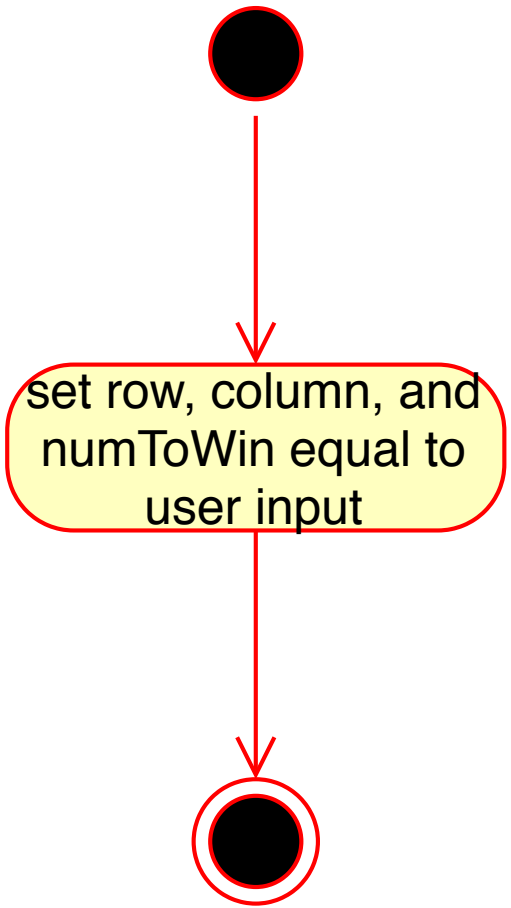
+ placeMarker(BoardPosition marker, char player) : void

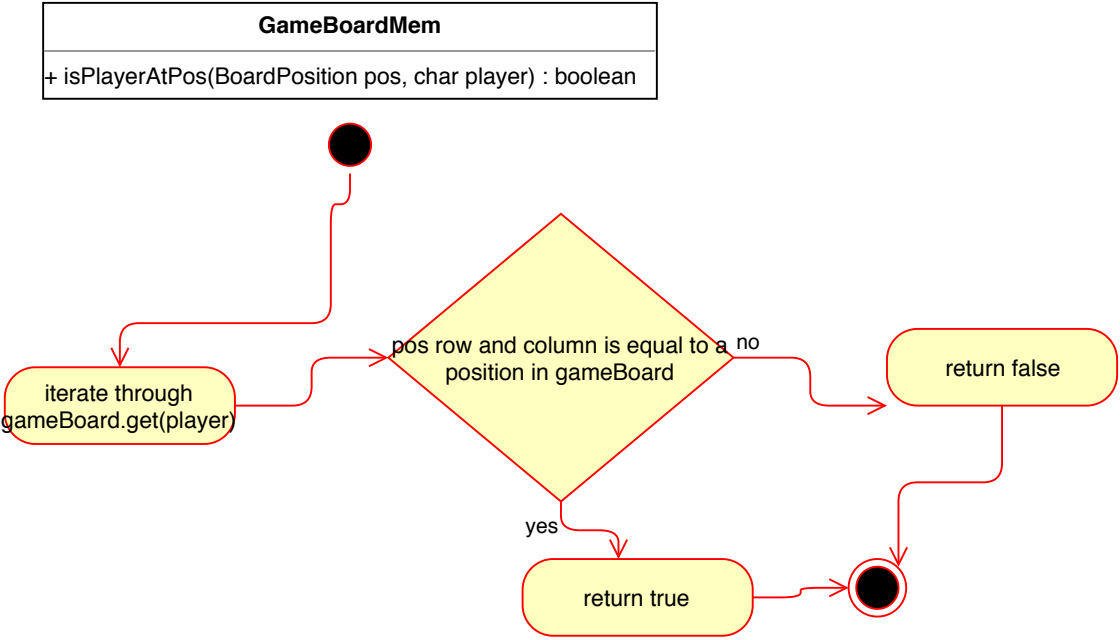


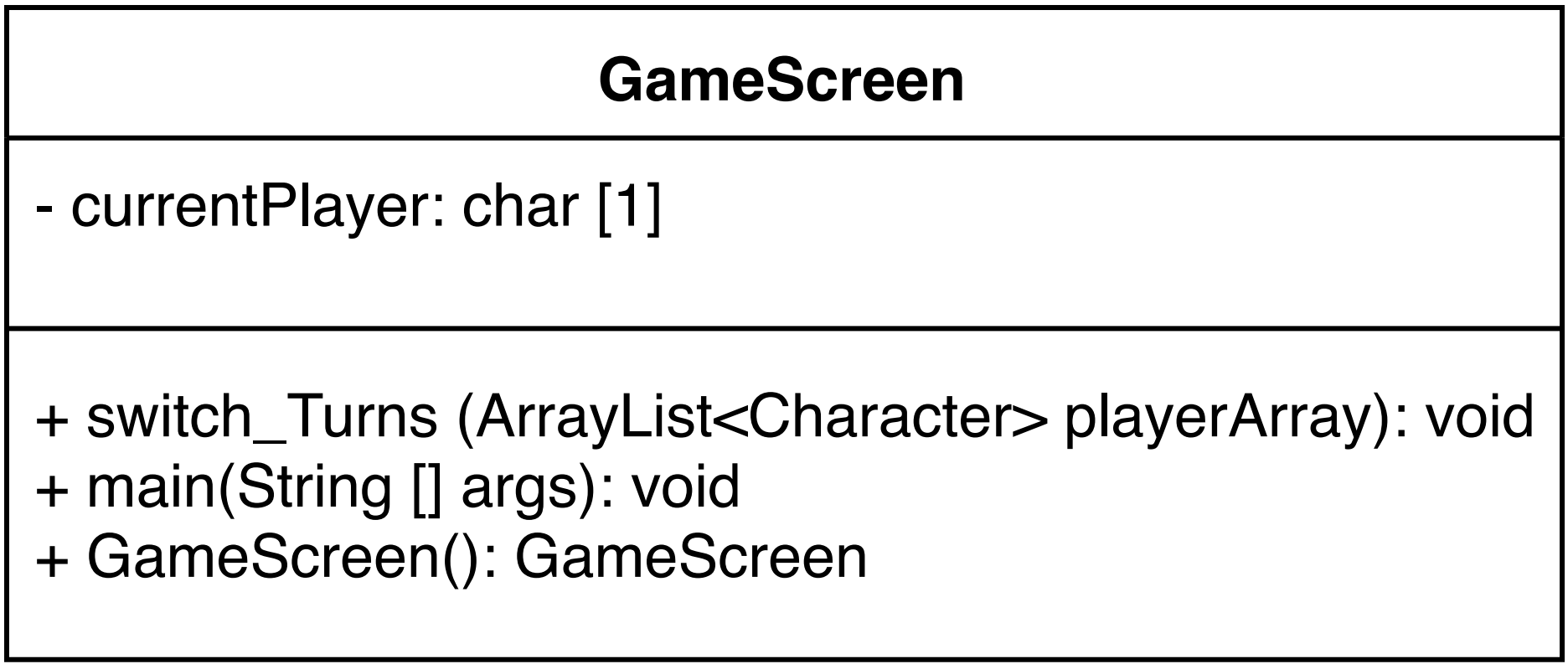

GameBoardMem
+ whatsAtPos(BoardPosition pos): char
iterate through gameBoard
find a position same as pos
no
return false
yes
return true

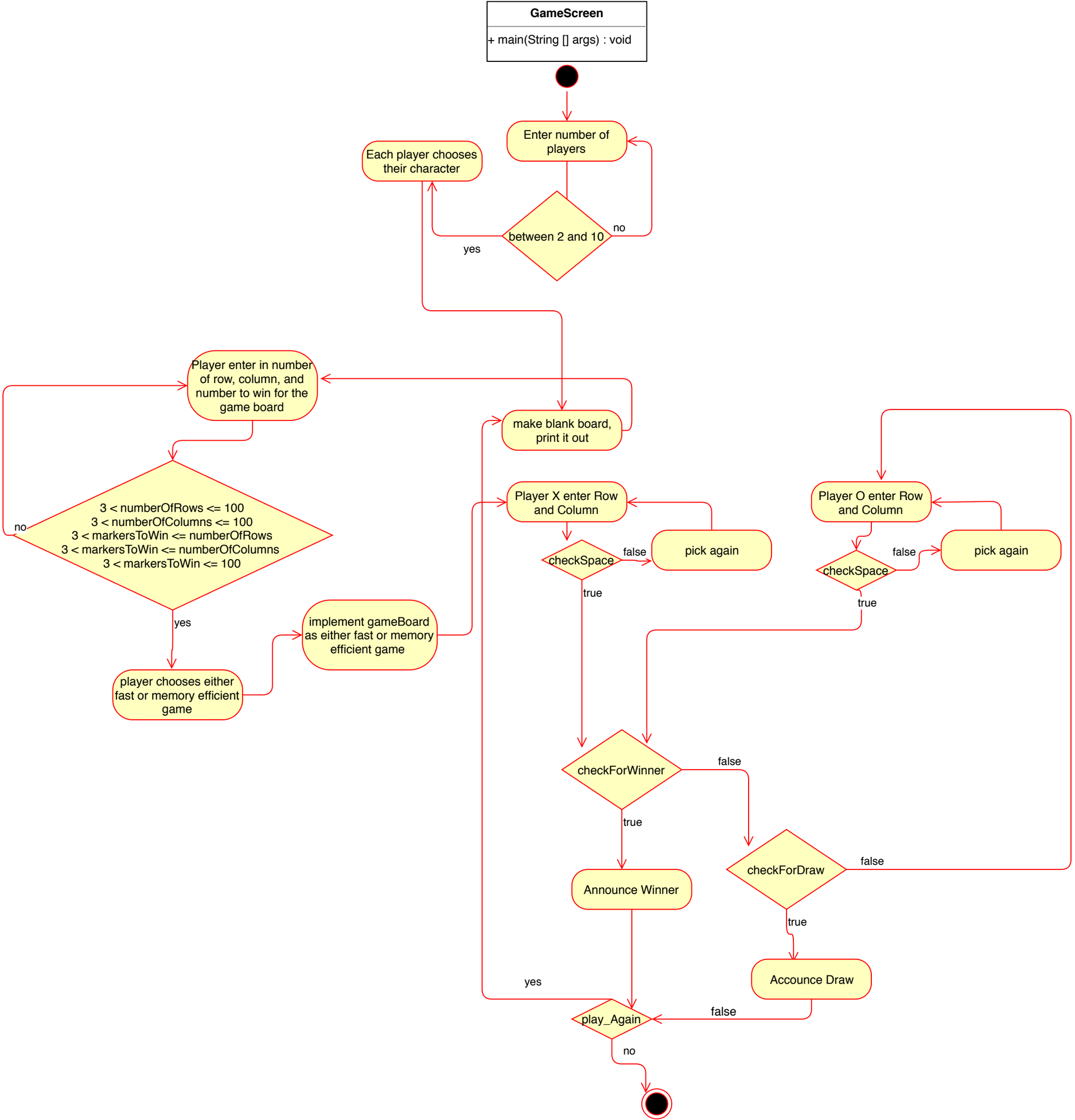
GameBoardMem

+ GameBoardMem(int rowInput, int columnInput, int numForWinInput): GameBoardMem



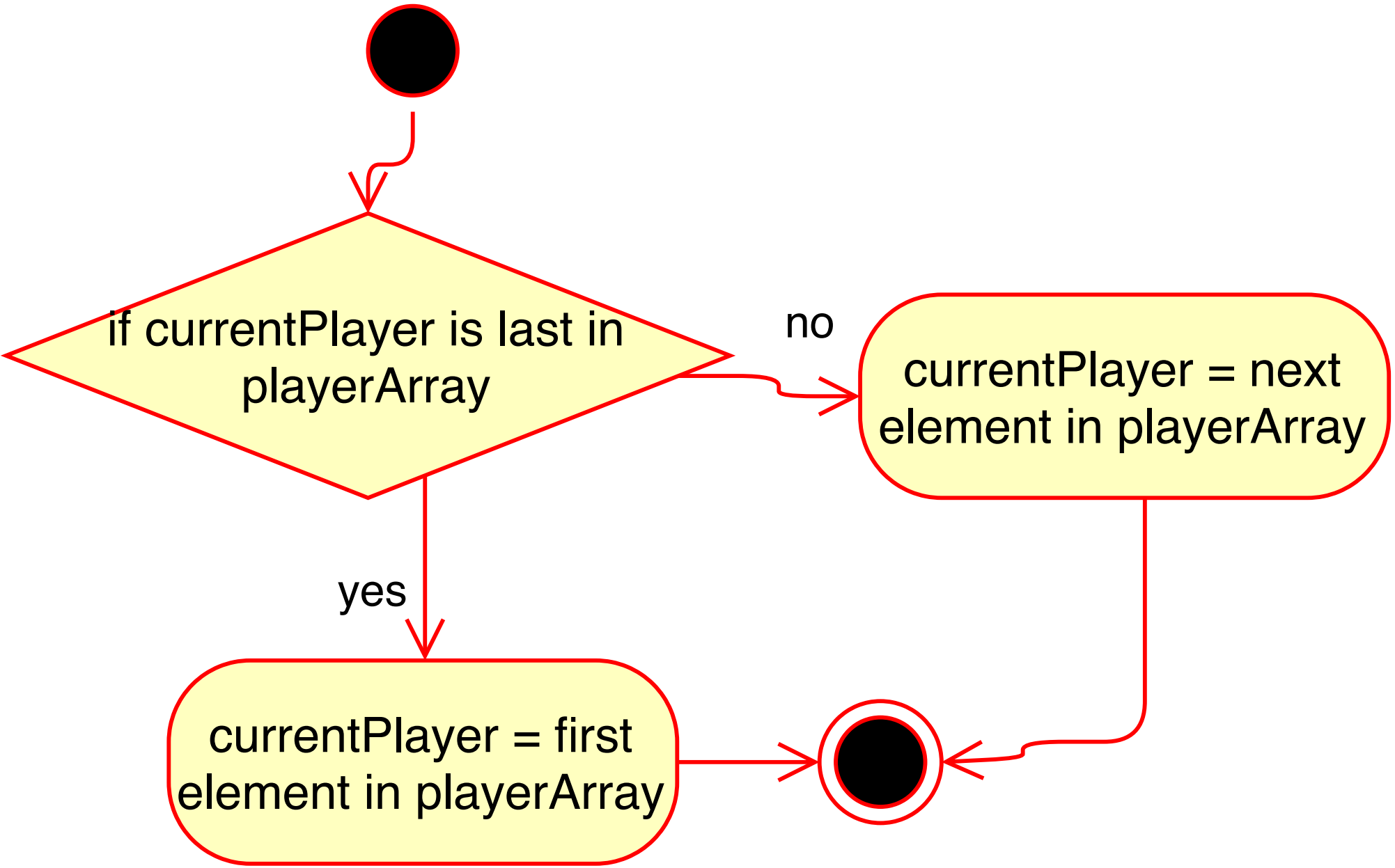

GameBoardMem
+ isPlayerAtPos(BoardPosition pos, char player) : boolean
iterate through gameBoard.get(player)
pos row and column is equal to a position in gameBoard
no
return false
yes
return true

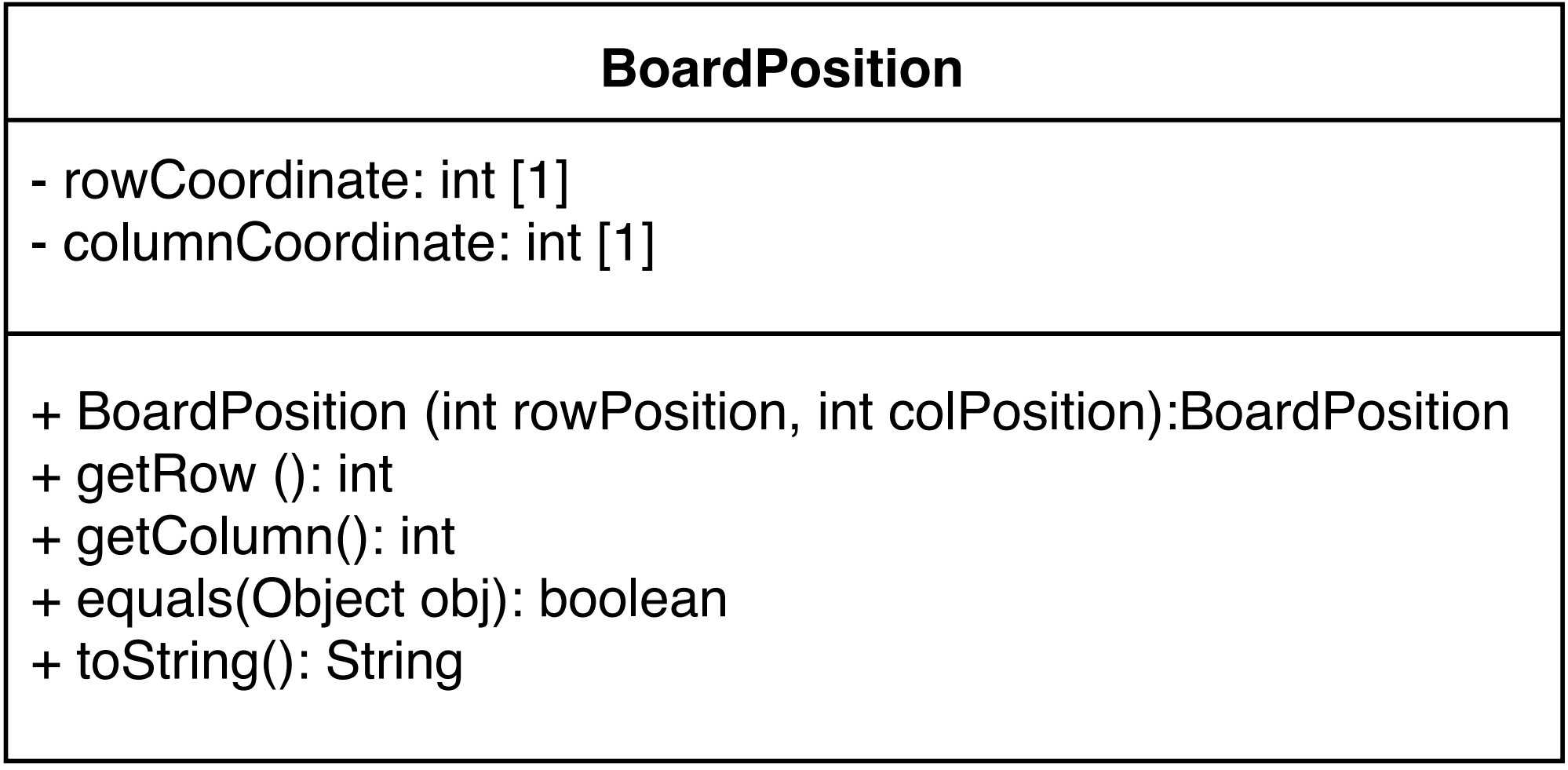

GameScreen
- currentPlayer: char [1]
+ switch_Turns (ArrayList<Character> playerArray): void
+ main(String [] args): void
+ GameScreen(): GameScreen

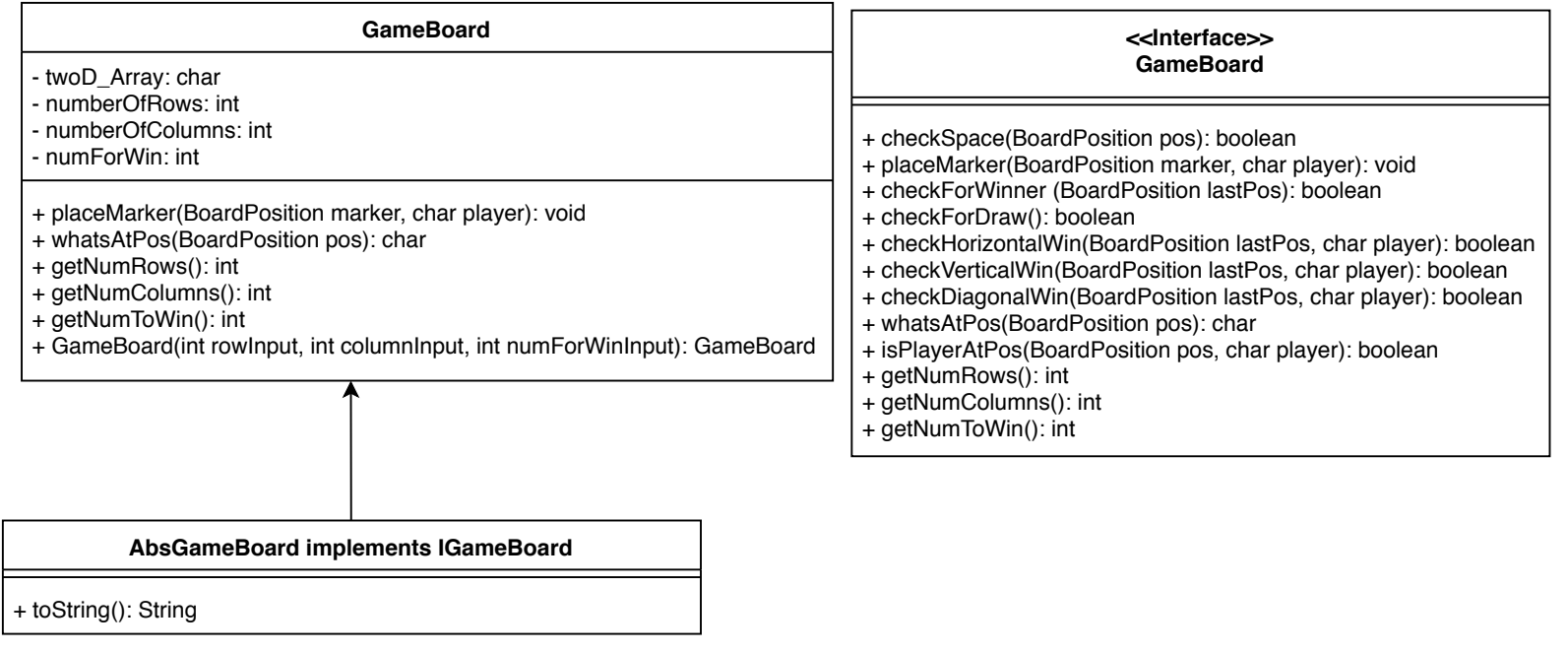

GameScreen
+ main(String [] args) : void
Enter number of players
between 2 and 10
no
yes
Each player chooses their character
Player enter in number of row, column, and number to win for the game board
3 < numberOfRows <= 100
3 < numberOfColumns <= 100
3 < markersToWin <= numberOfRows
3 < markersToWin <= numberOfColumns
3 < markersToWin <= 100
no
yes
player chooses either fast or memory efficient game
implement gameBoard as either fast or memory efficient game
make blank board, print it out
Player X enter Row and Column
checkSpace
false
pick again
true
Player O enter Row and Column
checkSpace
false
pick again
true
checkForWinner
false
true
Announce Winner
checkForDraw
false
true
Accounce Draw
play_Again
yes
false
no

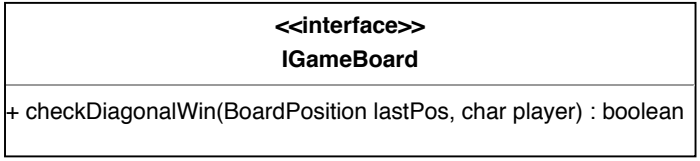
GameScreen

+ switch_Turns (ArrayList<Character> playerArray): void




BoardPosition
- rowCoordinate: int [1]
- columnCoordinate: int [1]
+ BoardPosition (int rowPosition, int colPosition):BoardPosition
+ getRow (): int
+ getColumn(): int
+ equals(Object obj): boolean
+ toString(): String


GameBoard
- twoD_Array: char
- numberOfRows: int
- numberOfColumns: int
- numForWin: int
+ placeMarker(BoardPosition marker, char player): void
+ whatsAtPos(BoardPosition pos): char
+ getNumRows(): int
+ getNumColumns(): int
+ getNumToWin(): int
+ GameBoard(int rowInput, int columnInput, int numForWinInput): GameBoard
<<Interface>>
GameBoard
+ checkSpace(BoardPosition pos): boolean
+ placeMarker(BoardPosition marker, char player): void
+ checkForWinner (BoardPosition lastPos): boolean
+ checkForDraw(): boolean
+ checkHorizontalWin(BoardPosition lastPos, char player): boolean
+ checkVerticalWin(BoardPosition lastPos, char player): boolean
+ checkDiagonalWin(BoardPosition lastPos, char player): boolean
+ whatsAtPos(BoardPosition pos): char
+ isPlayerAtPos(BoardPosition pos, char player): boolean
+ getNumRows(): int
+ getNumColumns(): int
+ getNumToWin(): int
AbsGameBoard implements IGameBoard
+ toString(): String



int tempRow = currentRow
int tempCol = currentColumn
count = 0
count2 = 0

if tempRow, tempCol equals current position and in bounds

no

yes

increase count by 1, tempRow++ tempCol++

if tempRow, tempCol equals current position and in bounds

no

yes

increase count by 1 tempRow-- tempCol--

if tempRow, tempCol equals current position and in bounds

yes

no

increase count2 by 1 tempRow+ tempCol--

if tempRow, tempCol equals current position and in bounds

yes

no

increase count2 by 1 tempRow-- tempCol++

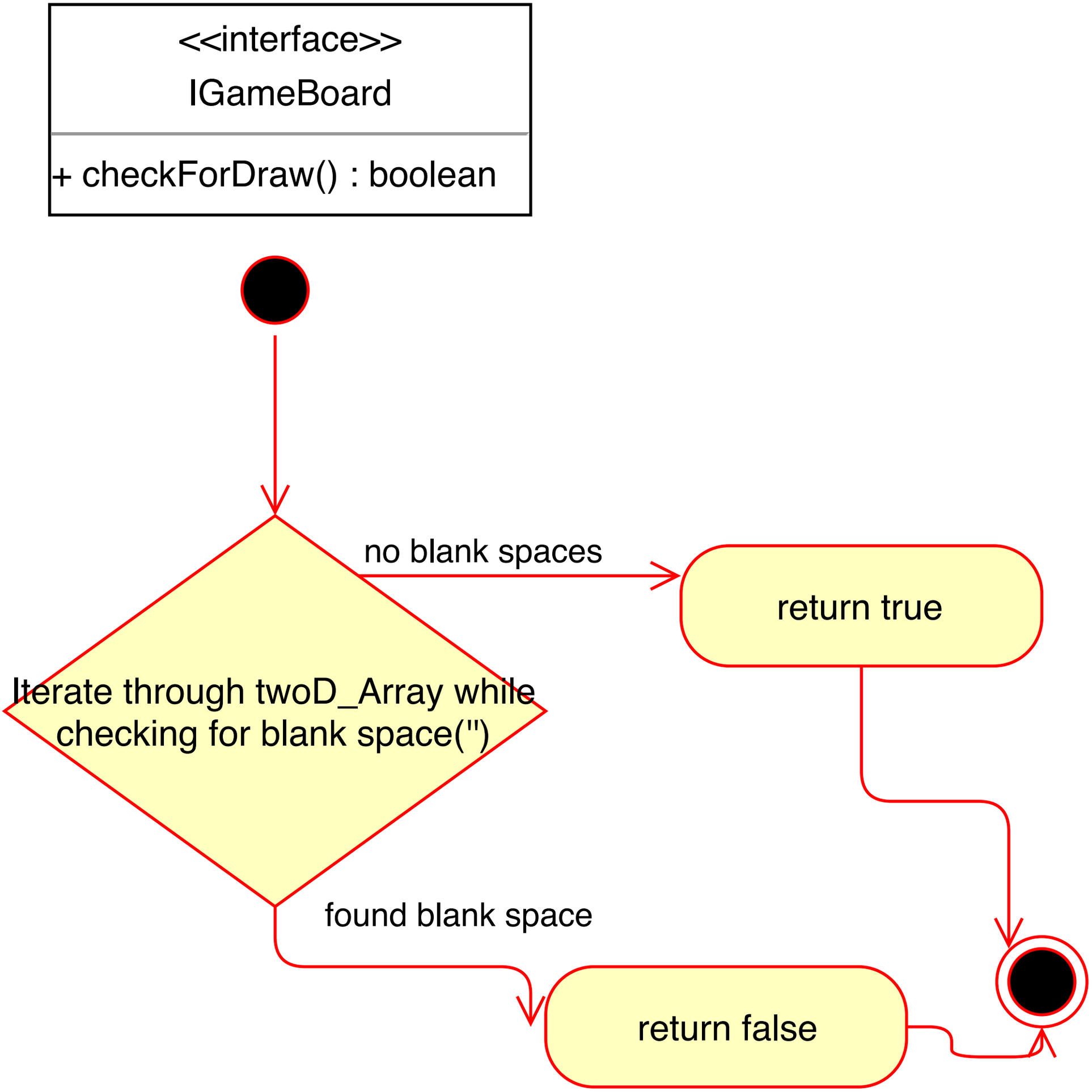
count == 5 or count2 == 5

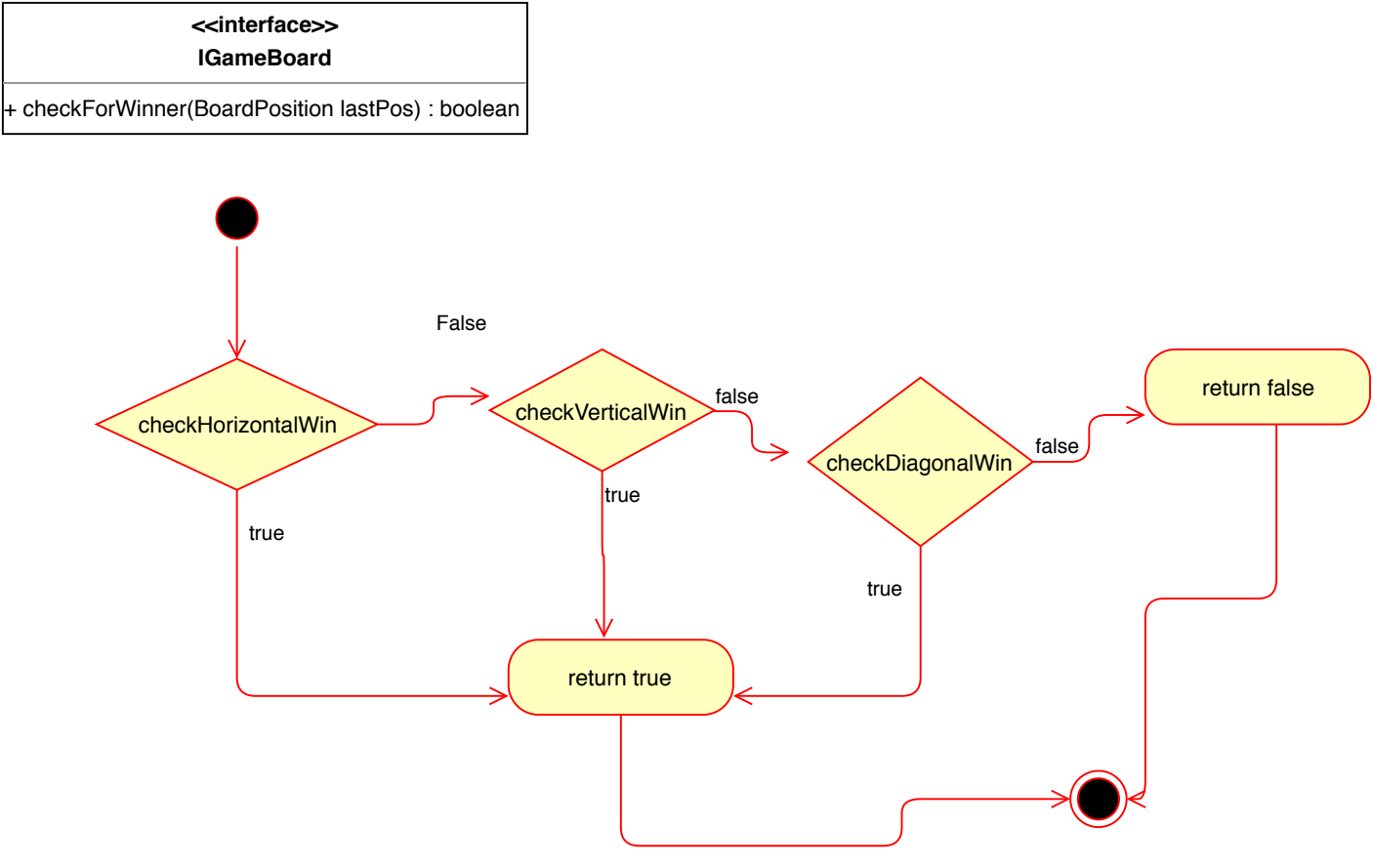
yes

no

return true

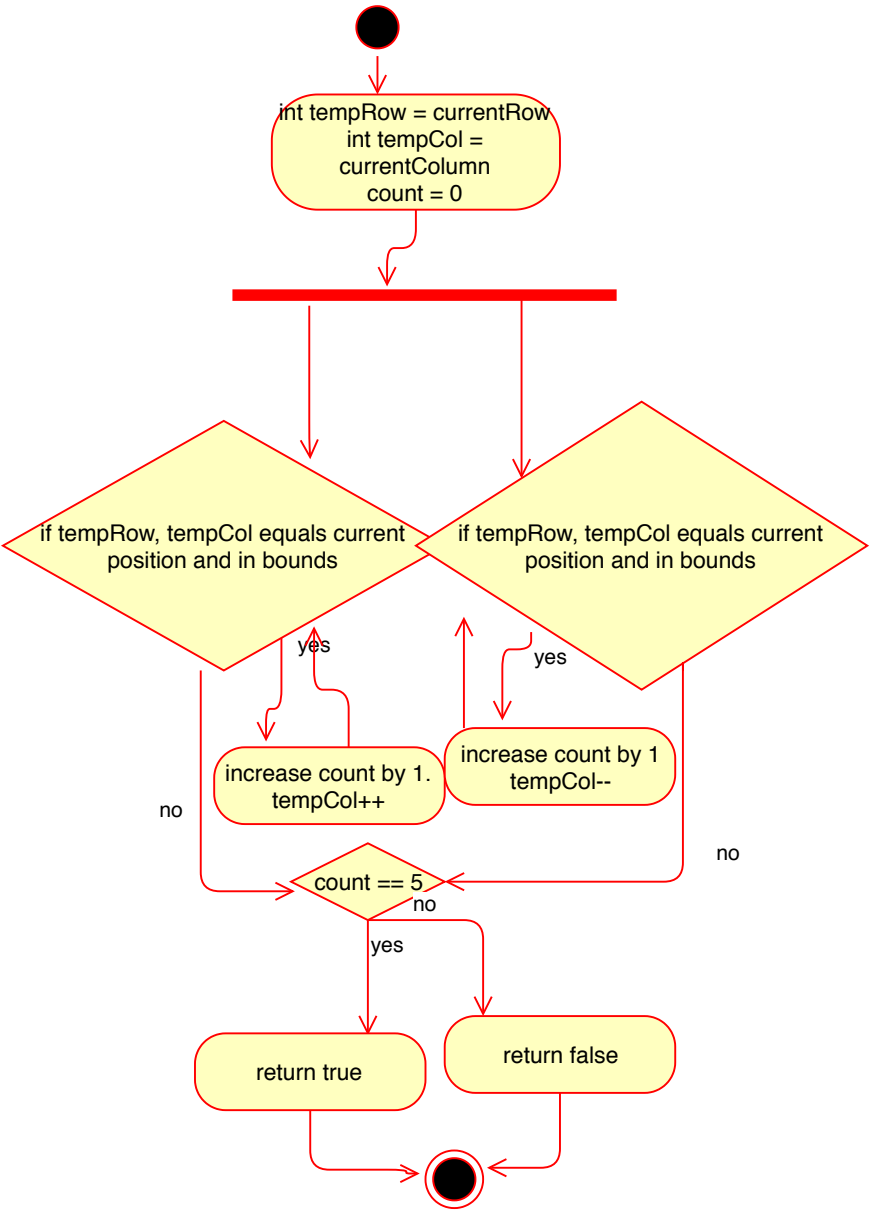
return false

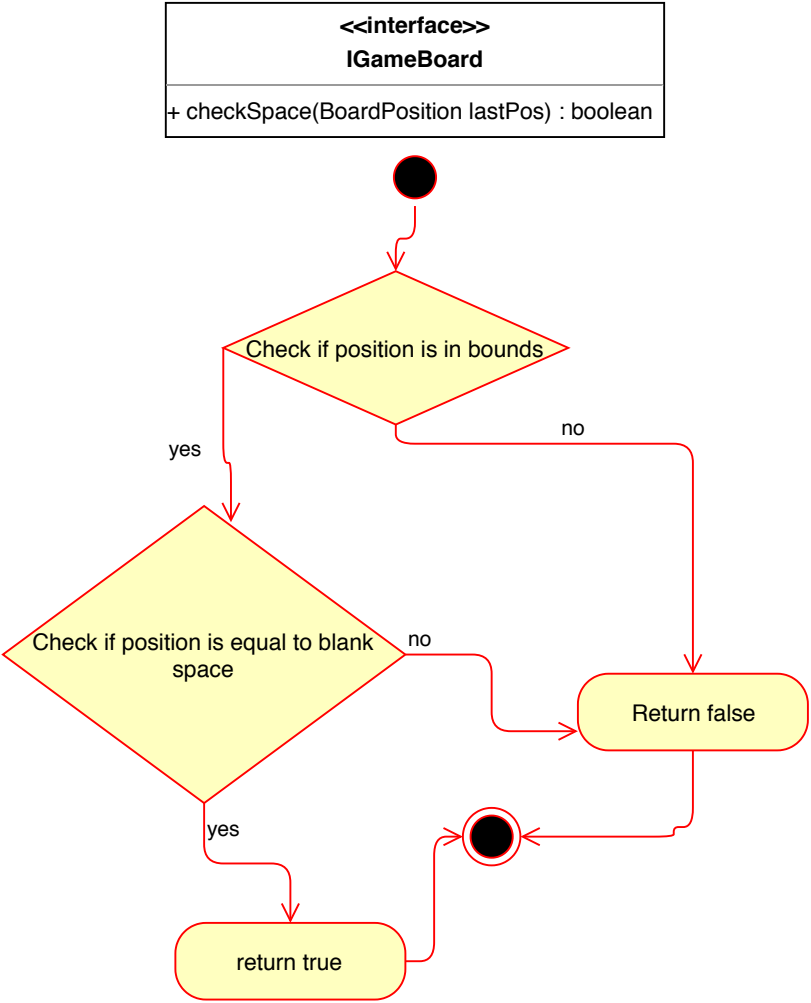

<<interface>>
IGameBoard
+ checkForDraw() : boolean
no blank spaces
return true
Iterate through twoD_Array while checking for blank space('')
found blank space
return false


<<interface>>
IGameBoard
+ checkForWinner(BoardPosition lastPos) : boolean
False
checkHorizontalWin
checkVerticalWin
checkDiagonalWin
return false
return true
false
false
true
true
true

<<interface>>
IGameBoard

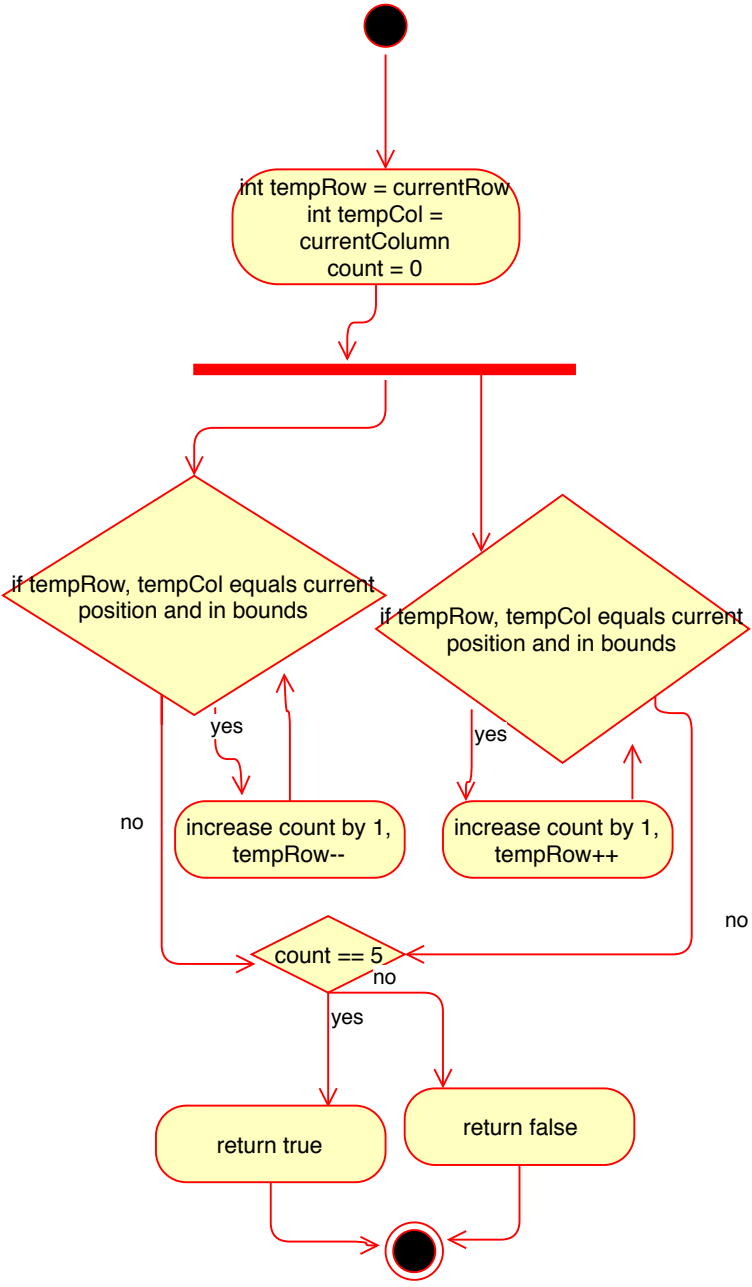
+ checkHorizontalWin(BoardPosition lastPos, char player) : boolean




<<interface>>
IGameBoard
+ checkSpace(BoardPosition lastPos) : boolean
Check if position is in bounds
yes
no
Check if position is equal to blank space
no
Return false
yes
return true

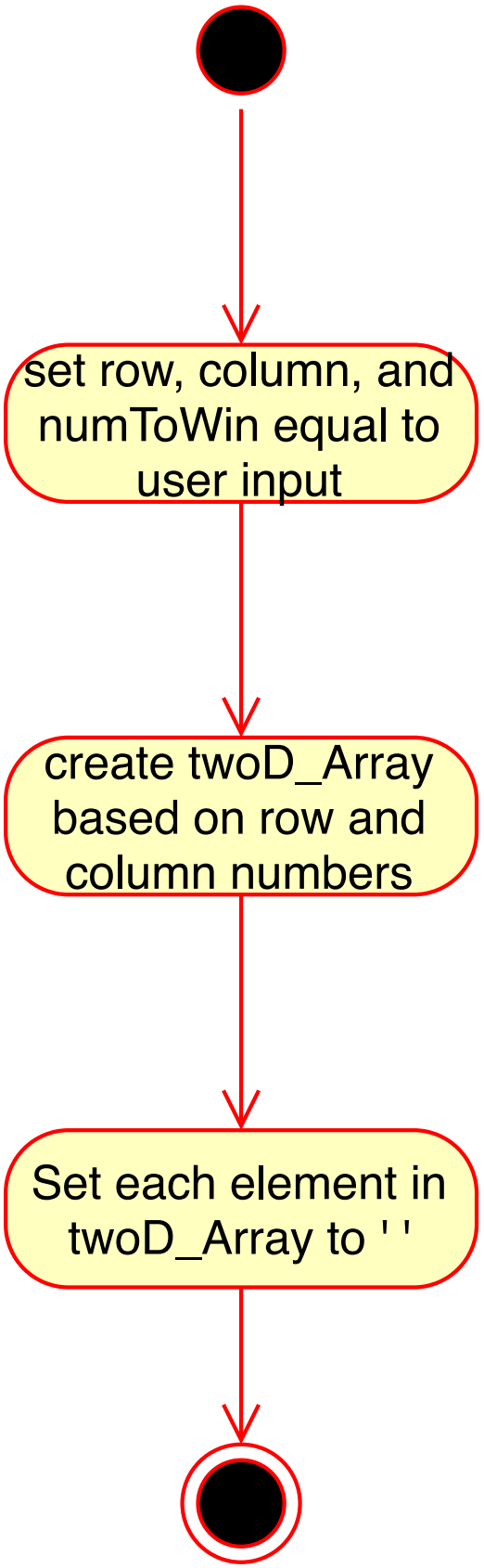
<<interface>>
IGameBoard

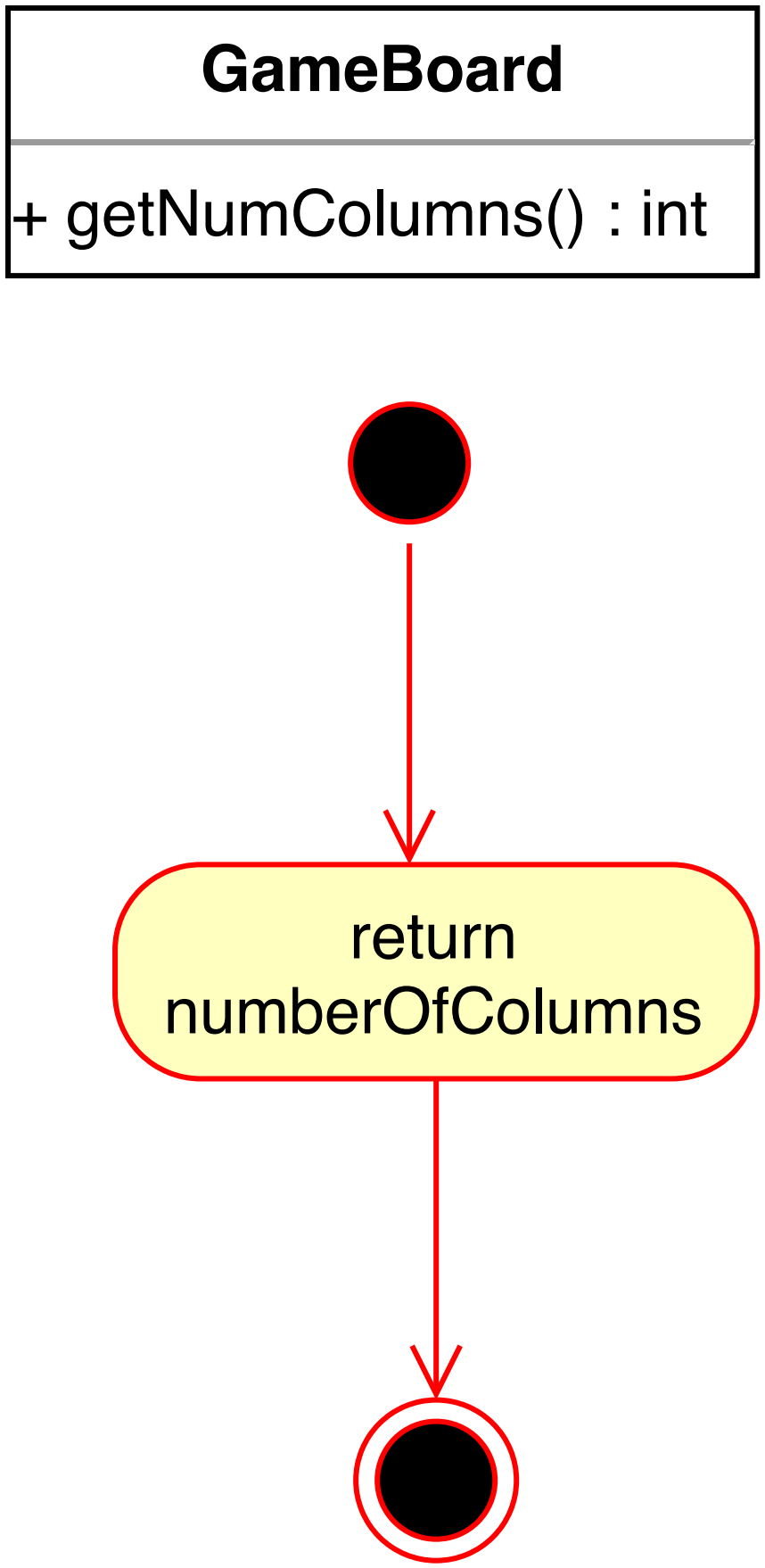
+ checkVerticalWin(BoardPosition lastPos, char player) : boolean

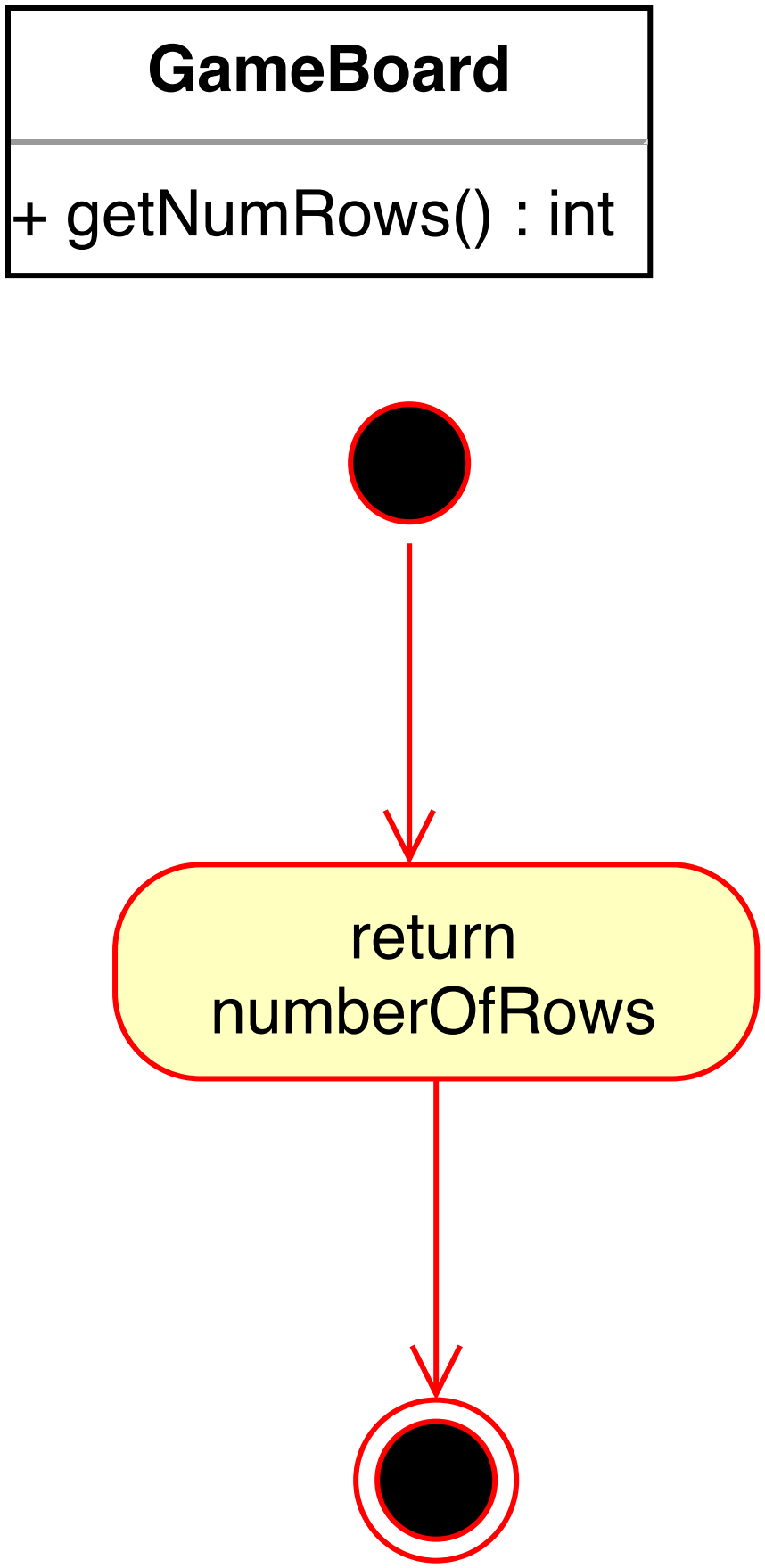


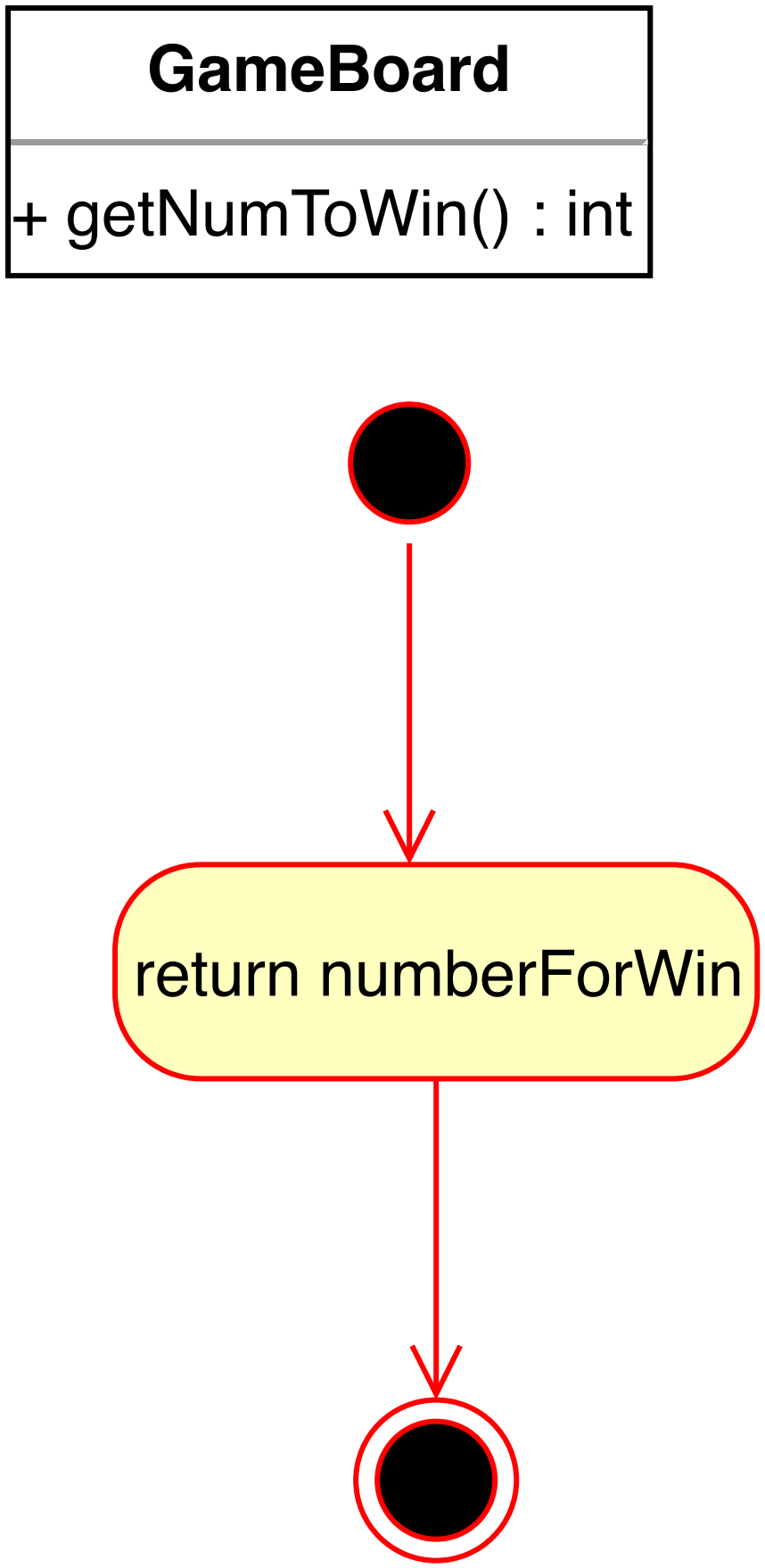
GameBoard

+ GameBoard(int rowInput, int columnInput, int numForWinInput): GameBoard



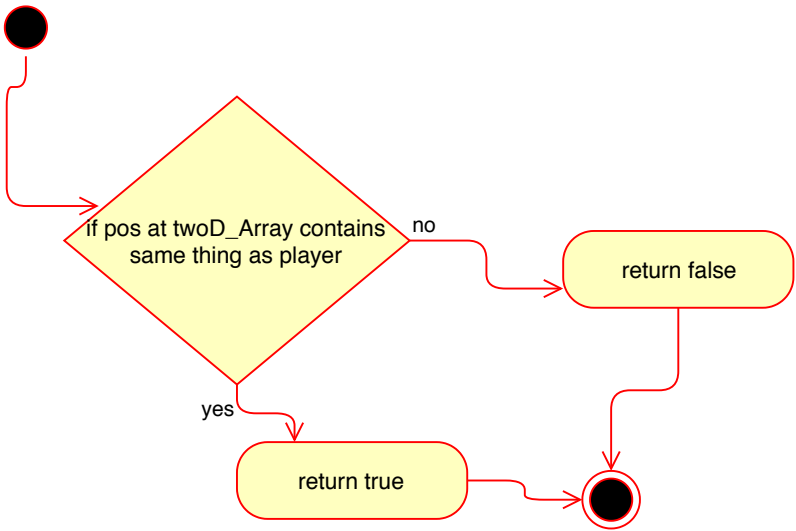

GameBoard
+ getNumColumns() : int
return
numberOfColumns


GameBoard
+ getNumRows() : int
return
numberOfRows


GameBoard
+ getNumToWin() : int
return numberForWin

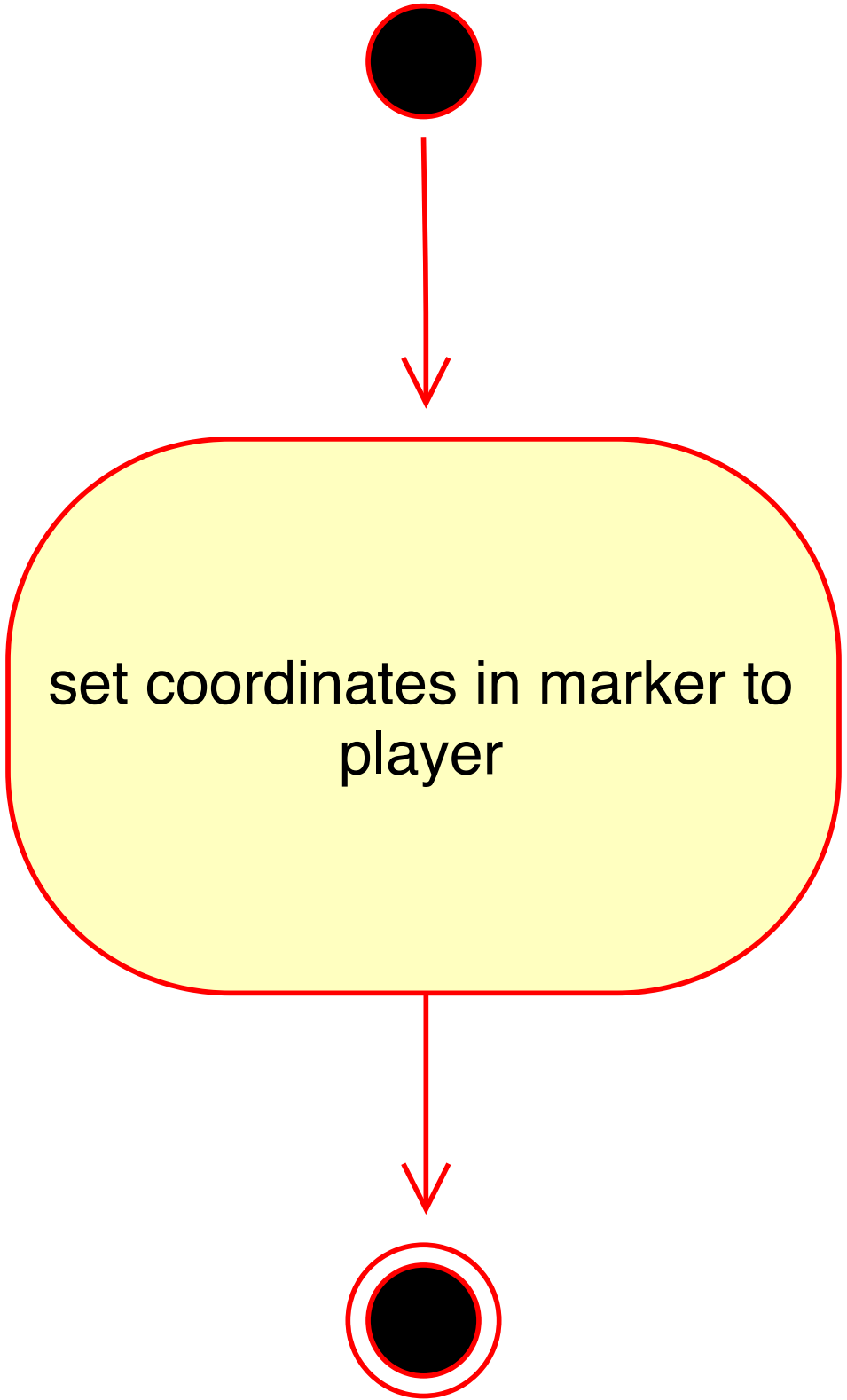
<<interface>>
IGameBoard

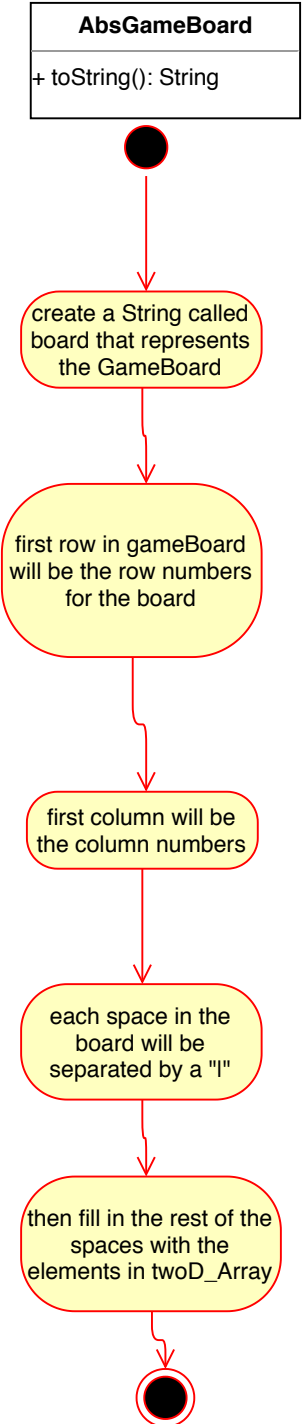
+ isPlayerAtPos(BoardPosition pos, char player) : boolean

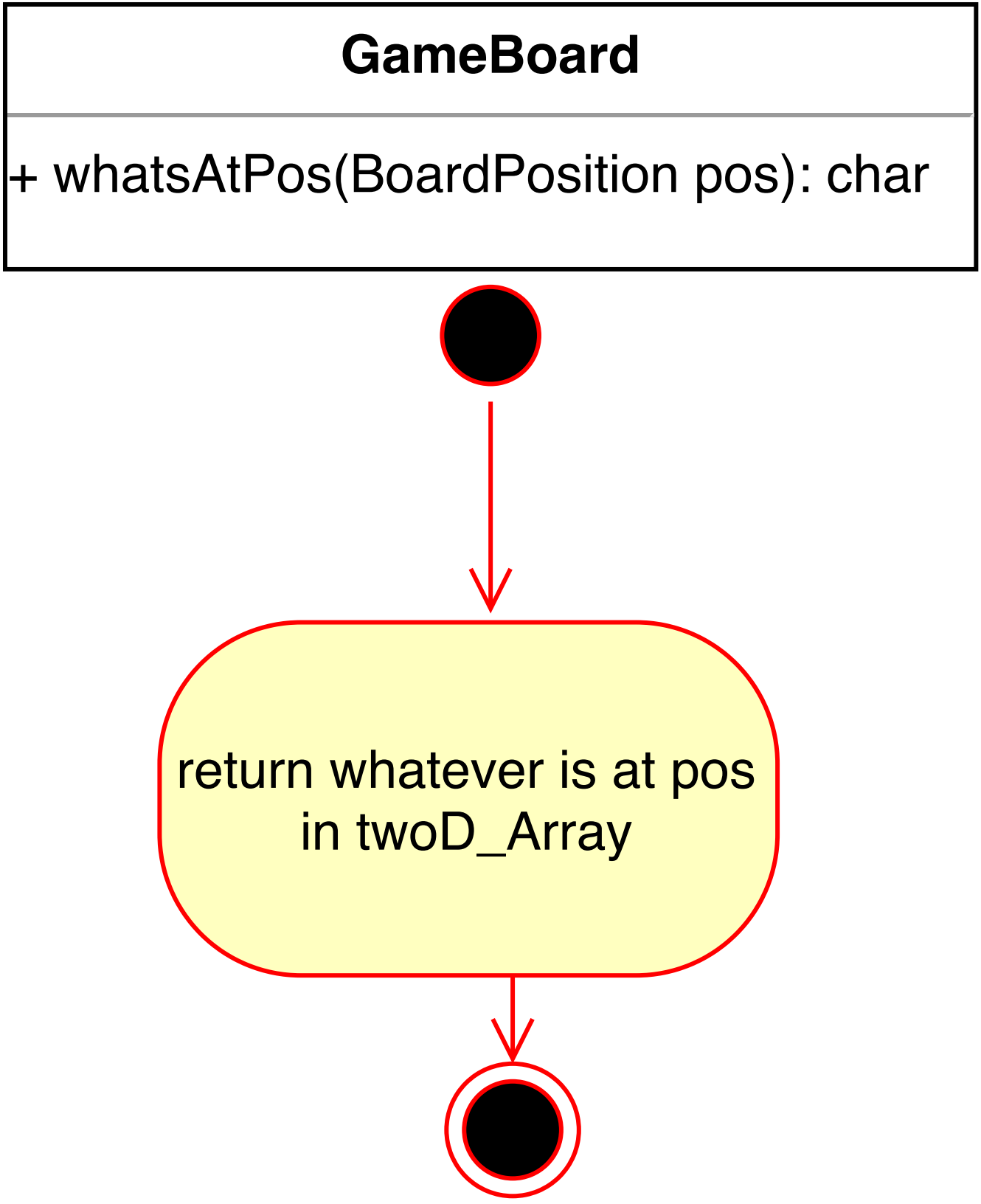


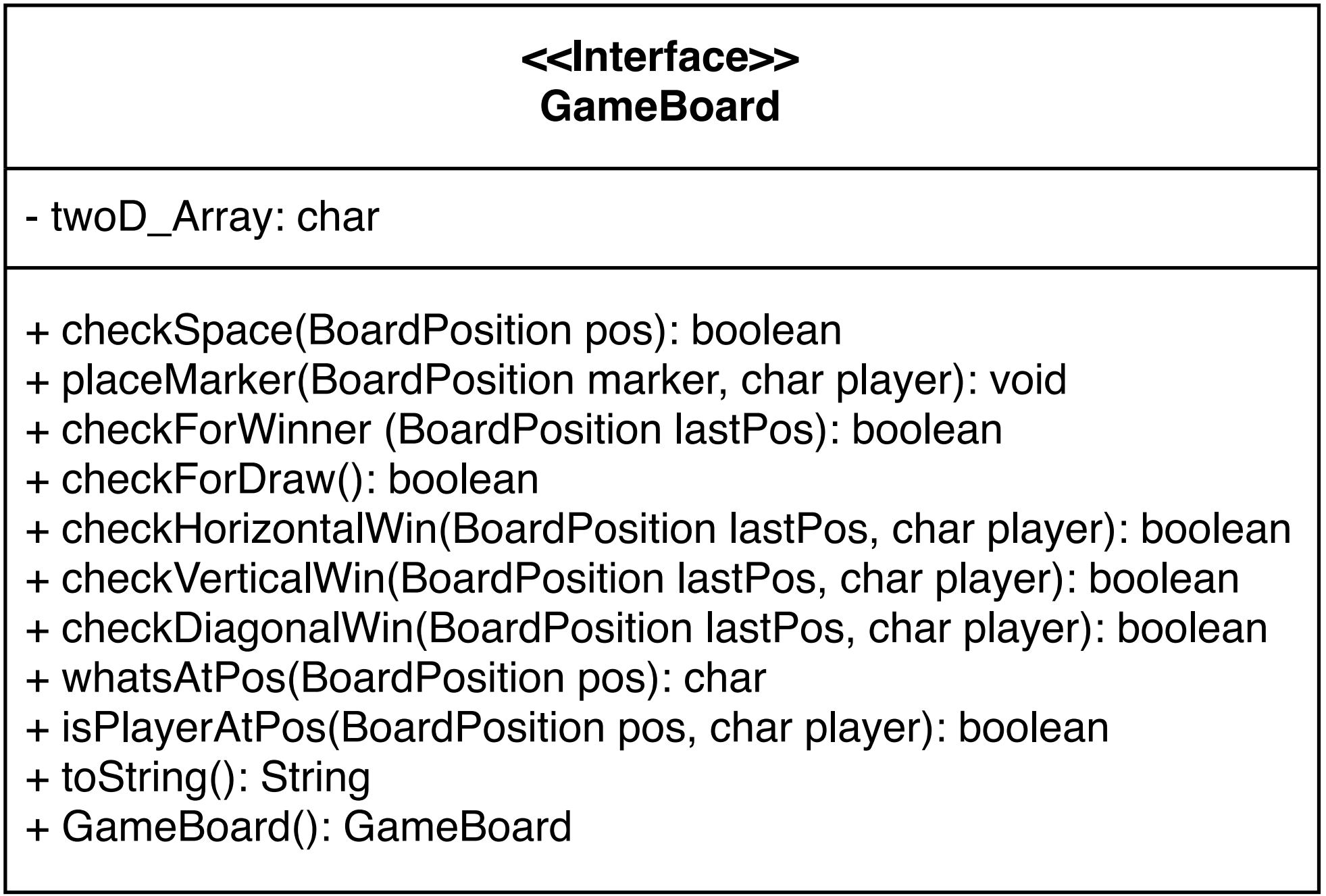
GameBoard

+ placeMarker(BoardPosition marker, char player) : void




AbsGameBoard
+ toString(): String
create a String called board that represents the GameBoard
first row in gameBoard will be the row numbers for the board
first column will be the column numbers
each space in the board will be separated by a "|"
then fill in the rest of the spaces with the elements in twoD_Array


GameBoard
+ whatsAtPos(BoardPosition pos): char
return whatever is at pos
in twoD_Array


<<Interface>>
GameBoard
- twoD_Array: char
+ checkSpace(BoardPosition pos): boolean
+ placeMarker(BoardPosition marker, char player): void
+ checkForWinner (BoardPosition lastPos): boolean
+ checkForDraw(): boolean
+ checkHorizontalWin(BoardPosition lastPos, char player): boolean
+ checkVerticalWin(BoardPosition lastPos, char player): boolean
+ checkDiagonalWin(BoardPosition lastPos, char player): boolean
+ whatsAtPos(BoardPosition pos): char
+ isPlayerAtPos(BoardPosition pos, char player): boolean
+ toString(): String
+ GameBoard(): GameBoard