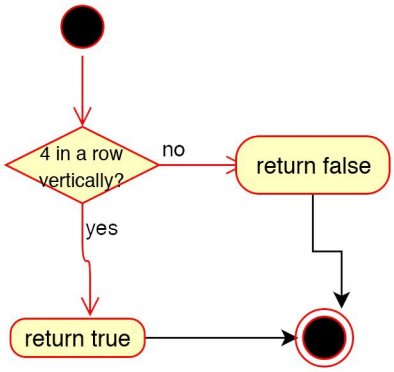
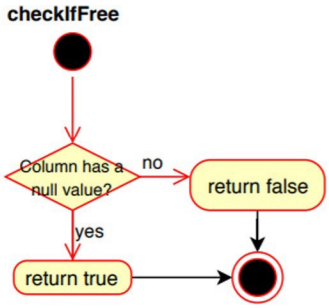
IGameBoard

+getNumRows(void):int
+getNumColumns(void):int
+getNumToWin(void):int
+ checkIfFree(int):boolean
+ checkForWin(int):boolean
+ placeToken(char, int):void
+ checkHorizWin(int, int, char):boolean
+ checkVertWin(int, int, char):boolean
+ checkDiagWin(int, int, char):boolean
+ whatsAtPos(int, int):char
+ checkTie(void):boolean

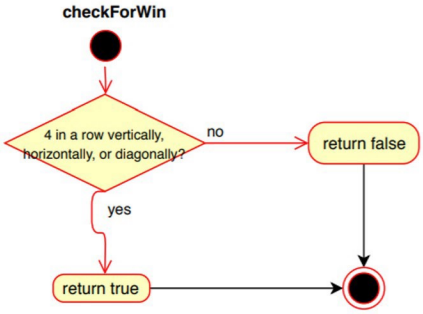
checkVertWin



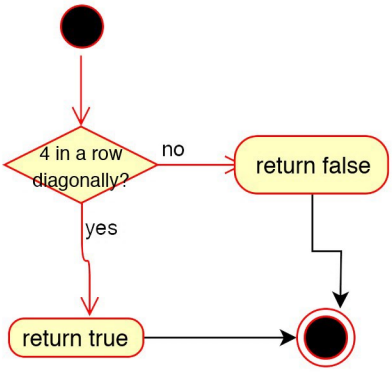
checkIfFree



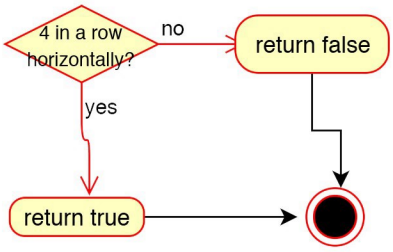
checkForWin



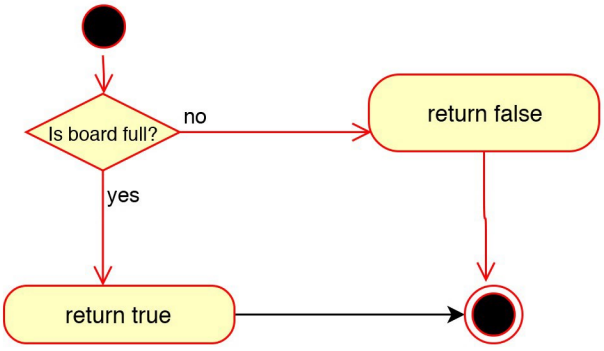
checkDiagWin

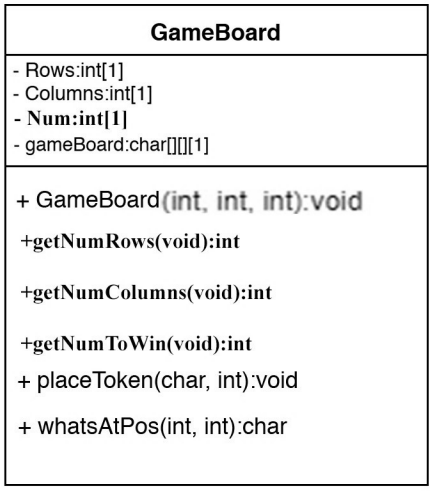


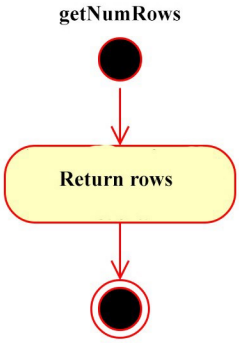
checkHorizWin

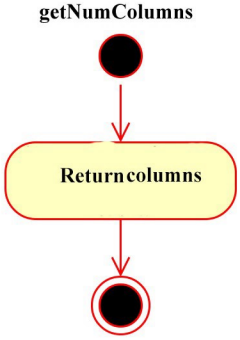


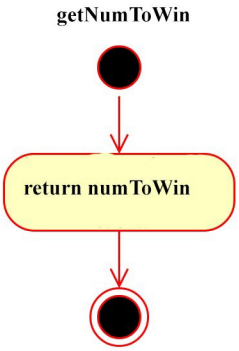
checkTie

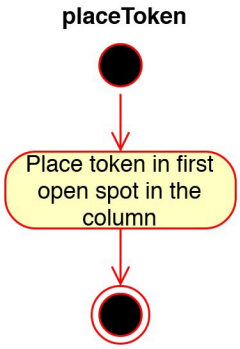


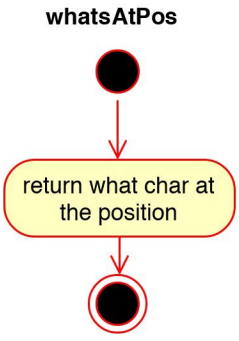

GameBoard
- Rows:int[1]
- Columns:int[1]
- Num:int[1]
- gameBoard:char[][][1]
+ GameBoard(int, int, int):void
+getNumRows(void):int
+getNumColumns(void):int
+getNumToWin(void):int
+ placeToken(char, int):void
+ whatsAtPos(int, int):char

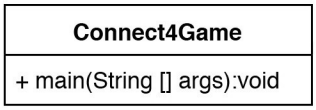

getNumRows
Return rows

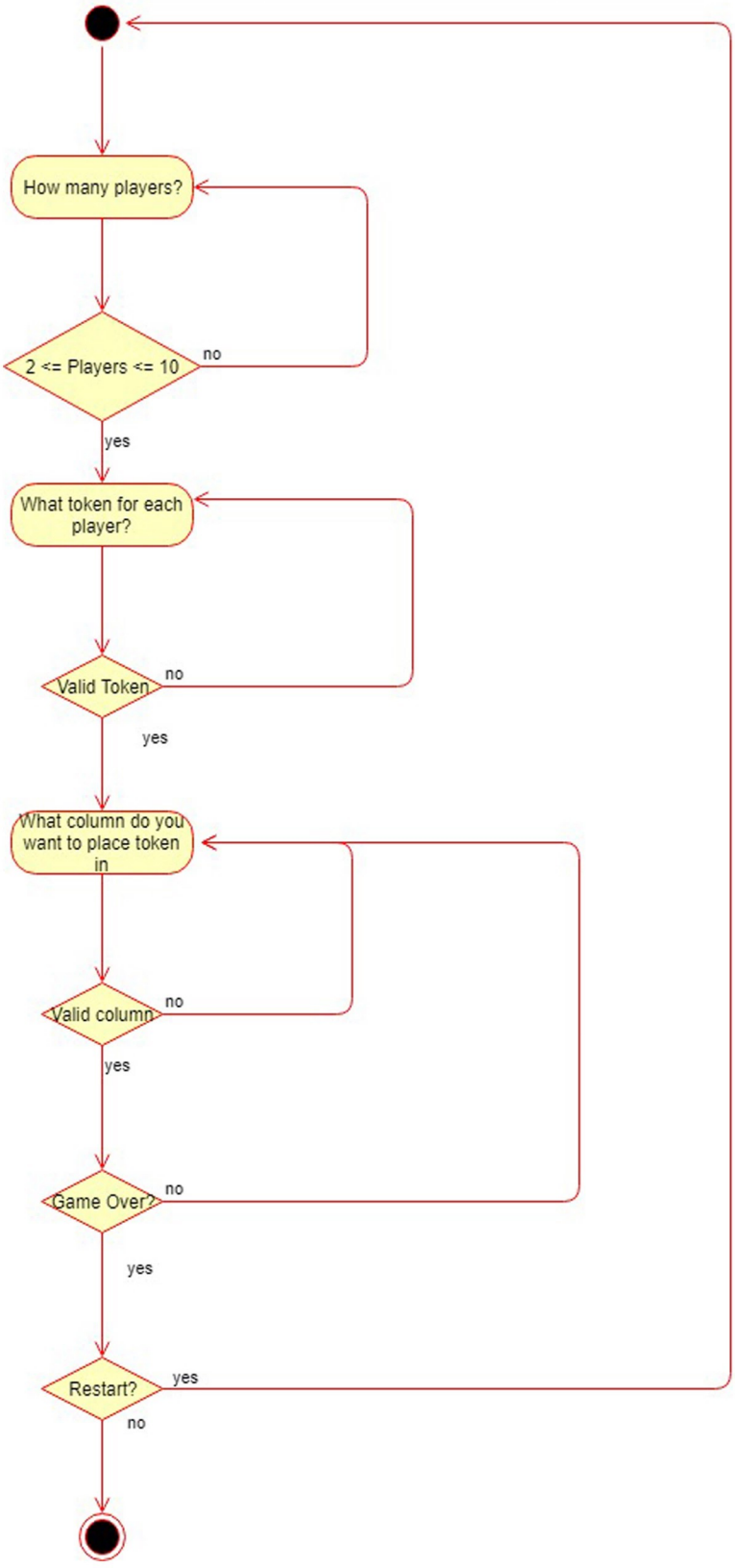

getNumColumns
Returncolumns


getNumToWin
return numToWin


placeToken
Place token in first open spot in the column


whatsAtPos
return what char at the position

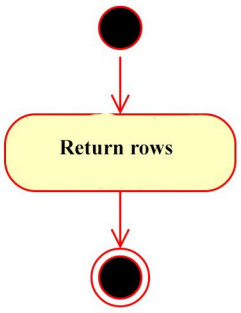

Connect4Game
+ main(String [] args):void


How many players?
2 <= Players <= 10
no
yes
What token for each player?
Valid Token
no
yes
What column do you want to place token in
Valid column
no
yes
Game Over?
no
yes
Restart?
yes
no

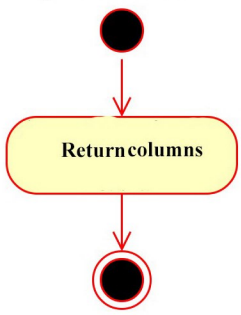
GameBoardMem
- gameMap: Map<Integer, List<Character>> - rows: int[1] - columns: int[1] - number: int[1]
+ GameBoardMem(int, int, int):void + getNumRows(void):int + getNumColumns(void):int + getNumToWin(void):int + placeToken(char, int):void + whatsAtPos(int, int):char

AbsGameBoard
+ toString(void): String

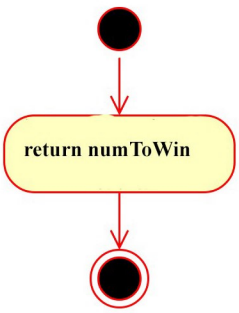
getNumRows



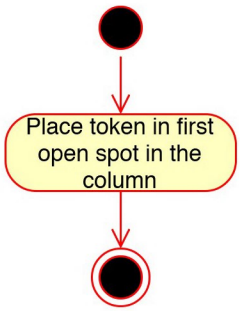
getNumColumns



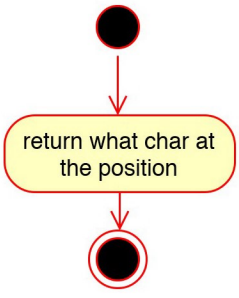
getNumToWin



placeToken



whatsAtPos



Functional Requirements

- Functional Requirement: User enters the number of players and their tokens

- Role: User

- Action: Enter number of players 2-10, and their respective unique tokens

- Benefit: Allows for up to 10 players to play together and have custom tokens

- User Story: As a user, I can play with friends and we can customize our game with our own tokens

---

- Functional Requirement: User enters the row, column, and number to win numbers for the game

- Role: User

- Action: Enter the number of rows 3-25, enter the number of columns 3-25, enter the number to win > 3

-Benefit: Allows user to customize board

- User Story: As a user, I can customize the game to be how the size I want, which is really nice

---

- Functional Requirement: User enters what column they want their token in

- Role: User

- Action: Enter column number 0-6

- Benefit: Allows them to play the game

- User Story: As a user, I can enter the column I want my token in to try and win the game and not lose a turn for bad input

---

- Functional Requirement: Program prints out the game board

- Role: User

- Action: View game board

- Benefit: Allows you to see what the previous moves are

- User Story: As a user I can view the game board to see previous moves and where I want to go next

---

- Functional Requirement: Program checks if there has been a win or a tie

- Role: User

- Action: See if there is a win/tie or keep playing

- Benefit: Allows you to know whether the game is over and if so who won

- User Story: The program outputs if there is a win or a tie, but if it doesn't I know to keep playing

---

Non-Functional Requirements

- System must be coded in Java
- System must run on Unix
- System must work for all possible inputs/situations of Connect 4

- Uses 2D Array / Hashmap

Deployment

Download the provided "hw4 .zip" file, and unzip it in your Unix system. Move into the hw4 directory where the Makefile is located, and from there you can type "make clean" in the terminal to remove unnecessary files, "make" to compile the program, and "make run" to run the program.