

Annie Hayes [ahayes5@g.clemson.edu](mailto:ahayes5@g.clemson.edu)

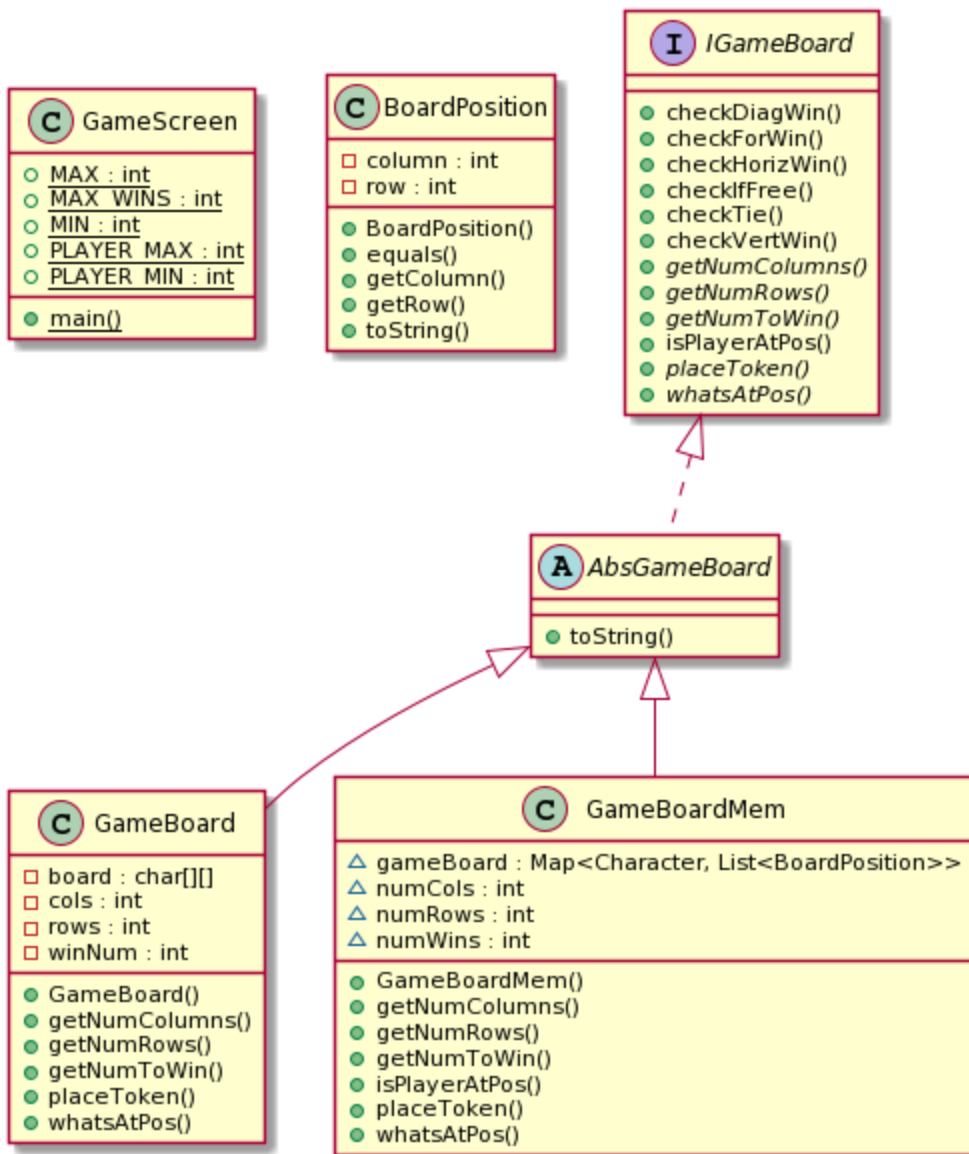
functional requirements:

1. As a player, I can only place a marker vertically in a column to simulate the connect 4 game.
2. As a user, I can only place a marker in the maximum set number of columns so I don't lose my turn
3. As a player, I cannot place a marker in a column that is already full so the board stays the same.
4. As a user, I can decide if I want to play a fast or memory-efficient game to save memory.
5. As a user, I can win if I have the user specific number of my markers in a row horizontally.
6. As a player, I can win if I have the user specific number of my markers in a row vertically.
7. As a player, I can win if I have the user specific number of my markers in a row diagonally.
8. As a strategy player, I can have up to 10 players play in my game.
9. As a connect-4 pro, I can have up to 100 columns and rows in my game to complicate the game for the players.
10. As a strategist, I can have a tie in the game because all of the columns are full.
11. As the user, I can alternate between players so each player can have a turn.
12. As a player, I can see whose turn it is so I know who is supposed to pick a column
13. As a connect-4 pro, I can pick which column to place my marker so I know which spot I played.
14. As a player, I cannot pick a spot outside of the bounds of the board or I will get an error message.
15. As a player, I can see if I have won by looking if I have the user specific number in a row.
16. As a player, I can play again once the game has ended.
17. As a player, I can enter an integer value to say which column I have selected.
18. As a player, I cannot enter a value over the user specific number since there are only a specific number of columns.
19. As a player, player 1 will start the game so it is consistent every game.
20. As a player, the board will keep track of all of the markers so I can see which positions are filled.

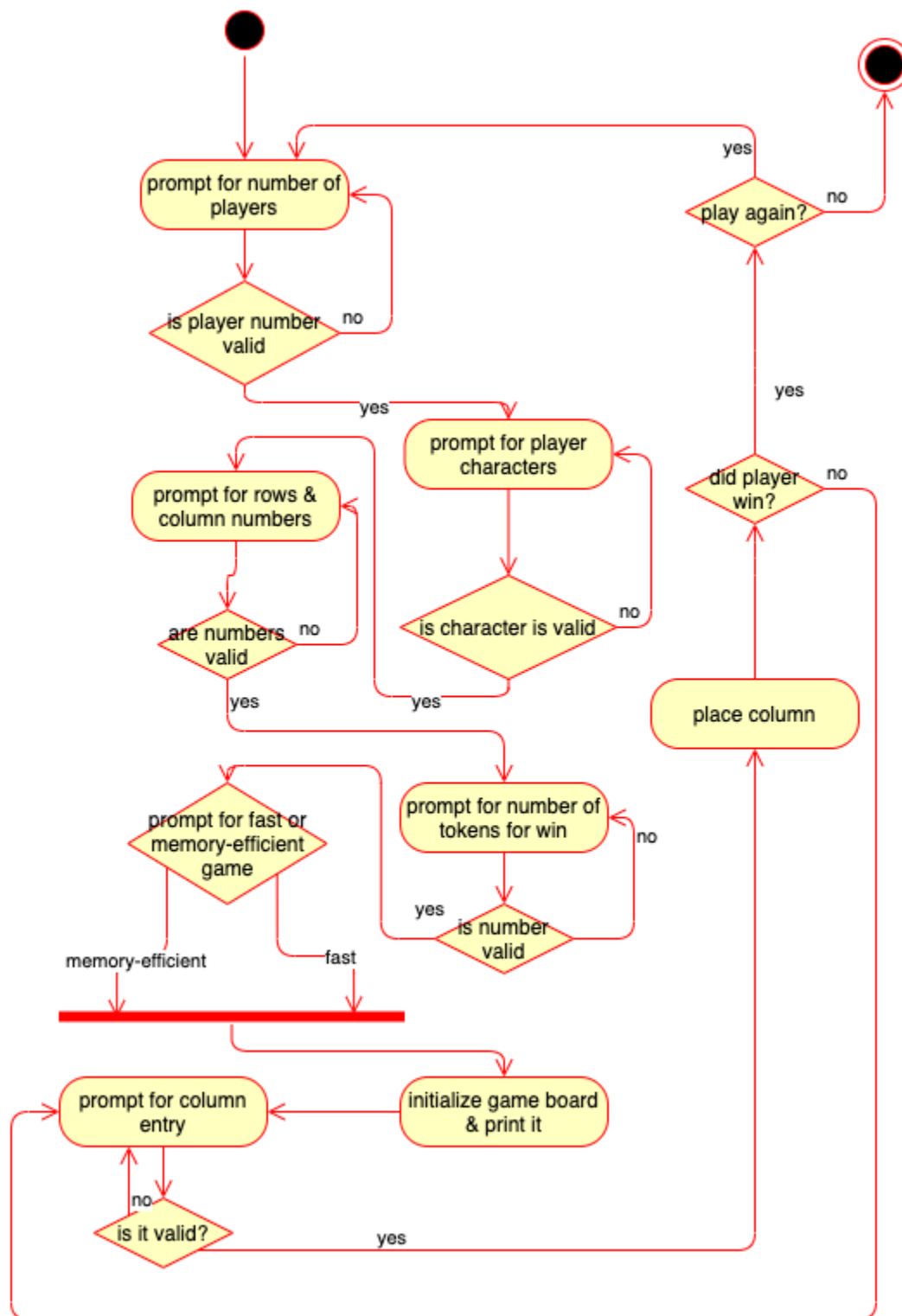
non-functional requirements

1. The system must be coded in Java
2. 0,0 is the bottom left corner of the board
3. game board size cannot exceed 100 x 100
4. player 1 goes first
5. The system must run on Unix
6. Do not use magic numbers
7. use good comments

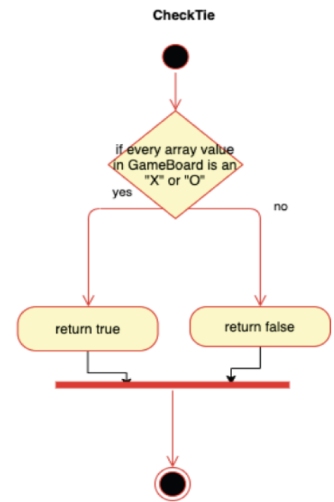
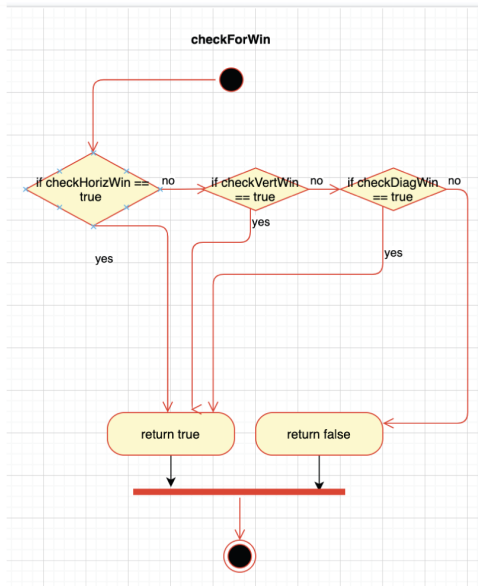
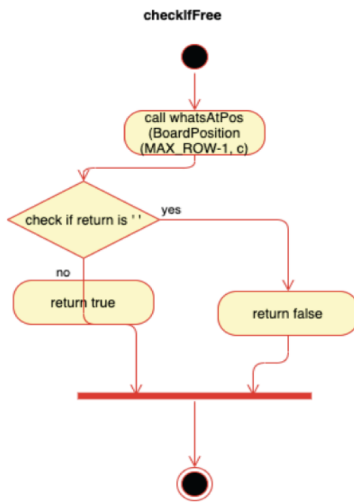
8. write contracts
9. make a program report
10. make UML class diagrams
11. make UML activity diagrams
12. write code for functions



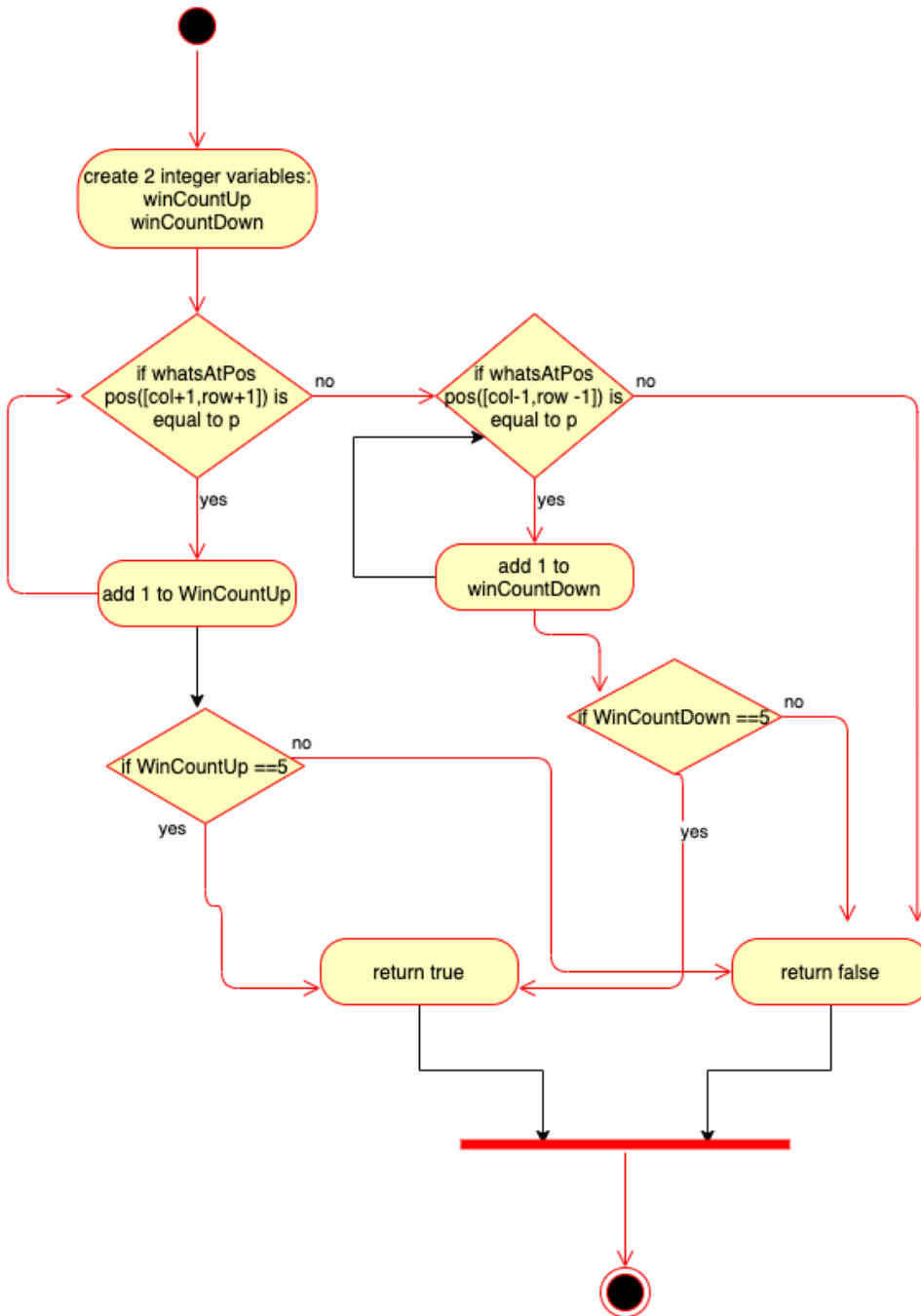
# GameScreen.java

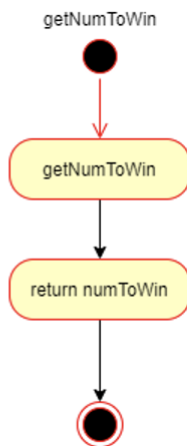
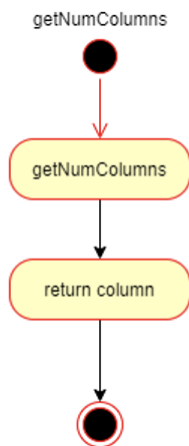
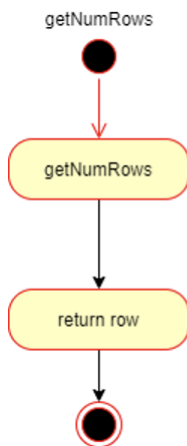
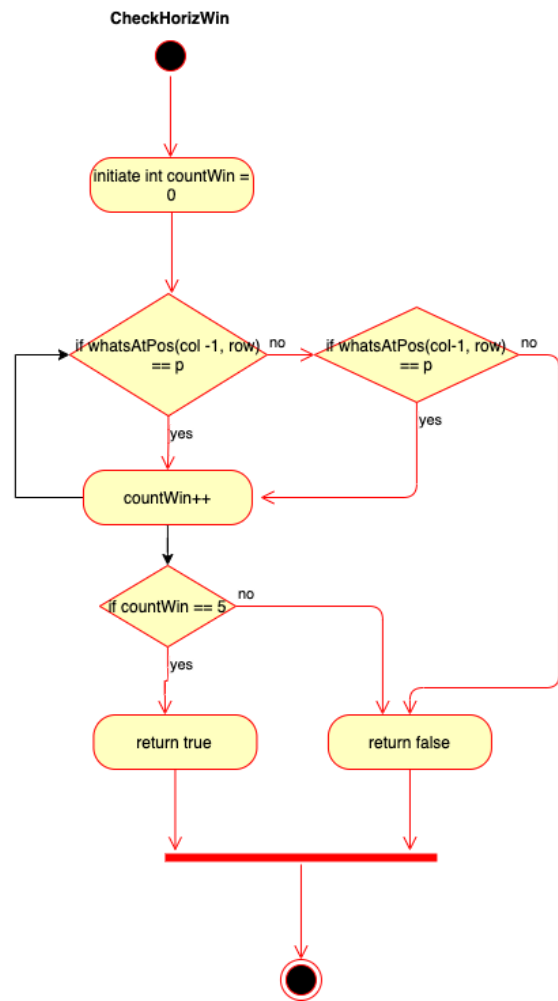
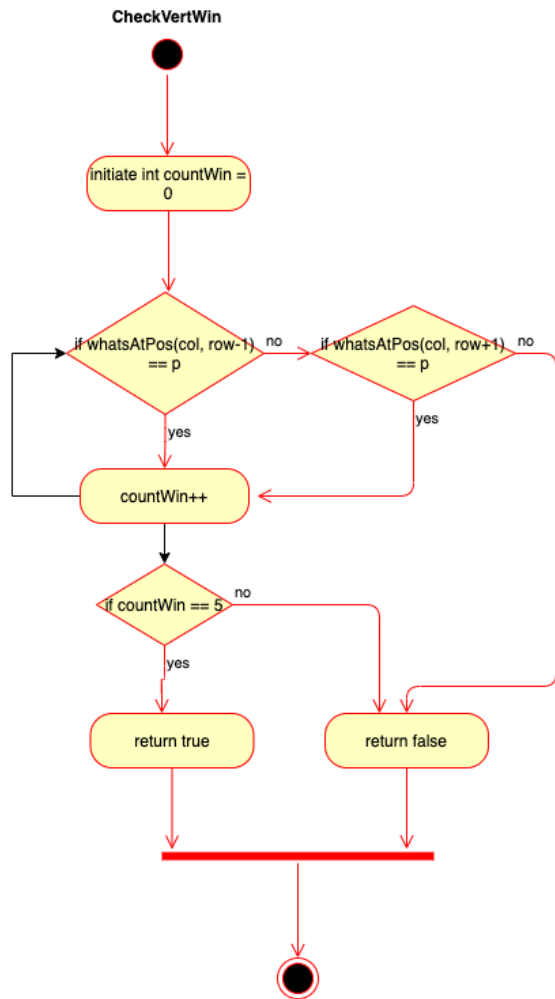


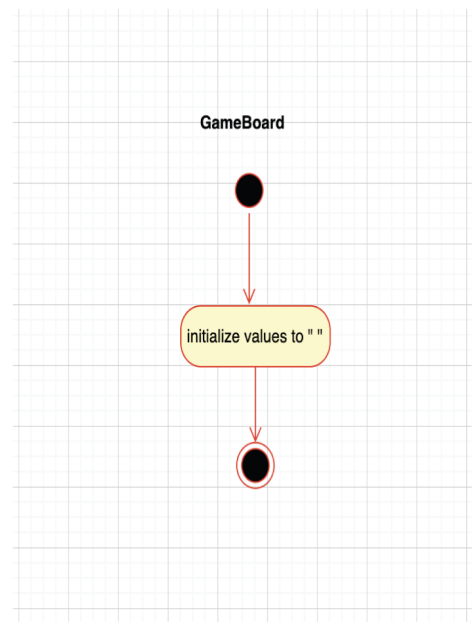
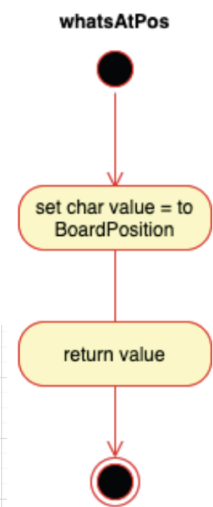
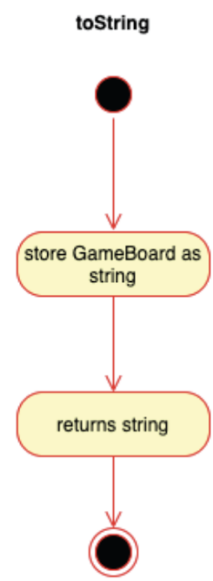
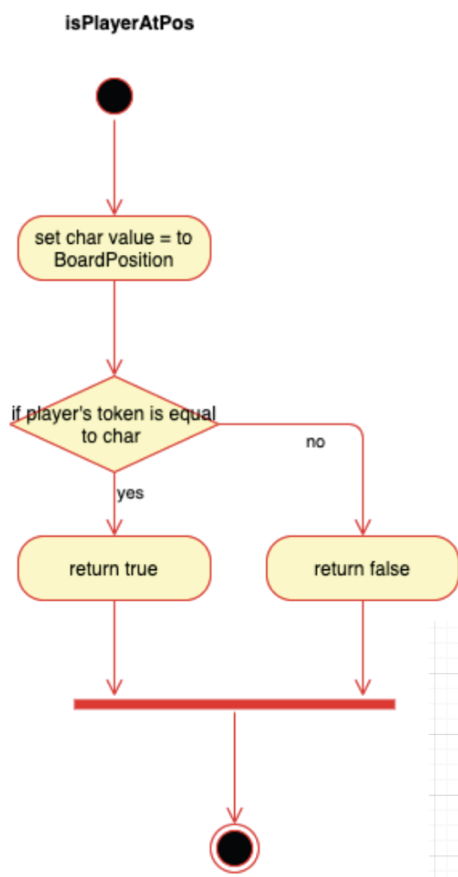
## GameBoard.java



# CheckDiagWin









## GameBoardMem.java

