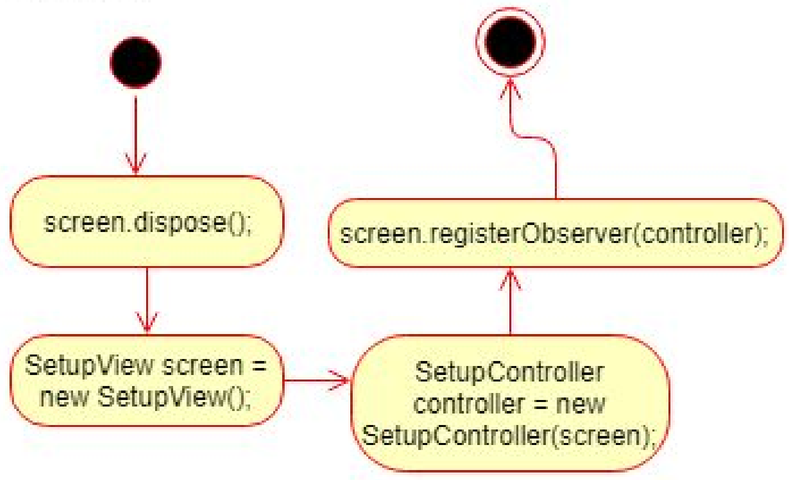
Requirements

- **Functional Requirements**
    - As a user, I can choose the number of rows in the game board, so that I can change the dimensions of the game board.
    - As a user, I can choose the number of columns in the game board, so that I can change the dimensions of the game board.
    - As a user, I can choose the number in a row needed to win, so that I can change the number in a row needed to win.
    - As a user, I can see the number in a row needed to win, so that I can know the number I need to reach in order to win the game.
    - As a user, I can select the number of players that I want to play with, so that I can play with up to nine other players.
    - As a user, I can select the character that I want to represent my player, so that I can distinguish my tokens from those of the other players.
    - As a user, I can choose if I want a fast game implementation or a memory efficient game implementation, so that I can choose whether I care more about saving memory or saving time.
    - As a user, I can choose to play again, so that I can try to win the game against my opponent again.
    - As a user, I can respecify the number of rows each time I play, so that I can change the dimensions of the game board each time.
    - As a user, I can respecify the number of columns each time I play, so that I can change the dimensions of the game board each time.
    - As a user, I can respecify the number in a row needed to win, so that I can know the number I need to reach in order to win the game.
    - As a user, I can respecify whether I want a fast or memory efficient game implementation, so that I can choose this specification for each individual game.
    - As a user, I can respecify the number of players each time I play, so that I can play with a different number of players each time.
    - As a user, I can see when my opponent or I has won the game, so that I can know if the game is over.
    - As a user, I can see when the game has tied, so that I can know if the game is over.
    - As a user, I can see if my input was invalid (out of bounds), so that I can know to choose a different row to place my token.
    - As a user, I can see if my input was invalid (column was full), so that I can know to choose a different row to place my token.
    - As a user, I can try input that may be invalid without losing my turn, so that I can continue to play the game.
- **Non-Functional Requirements**
    - A new game should always start with Player X
    - Code should be well formatted and documented
    - Code should have good, accurate contracts
    - Program should be easy to use, read, and understand
    - Code should follow Design By Contract and the Model View Controller Design Pattern
    - Code should follow programming to the interface
    - Game should use the provided Graphical User Interface

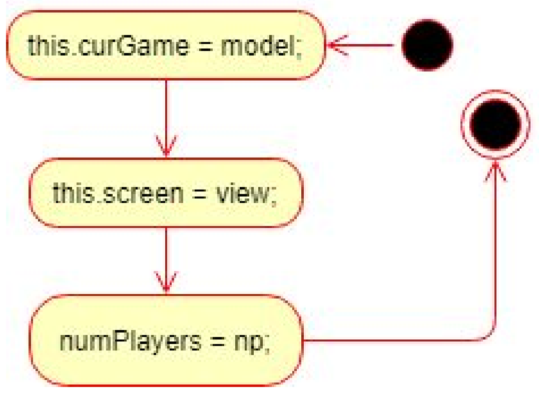
Design

- ConnectXController

newGame()


screen.dispose();
SetupView screen = new SetupView();
SetupController controller = new SetupController(screen);
screen.registerObserver(controller);

ConnectXController(IGameBoard model, ConnectXView view, int np)

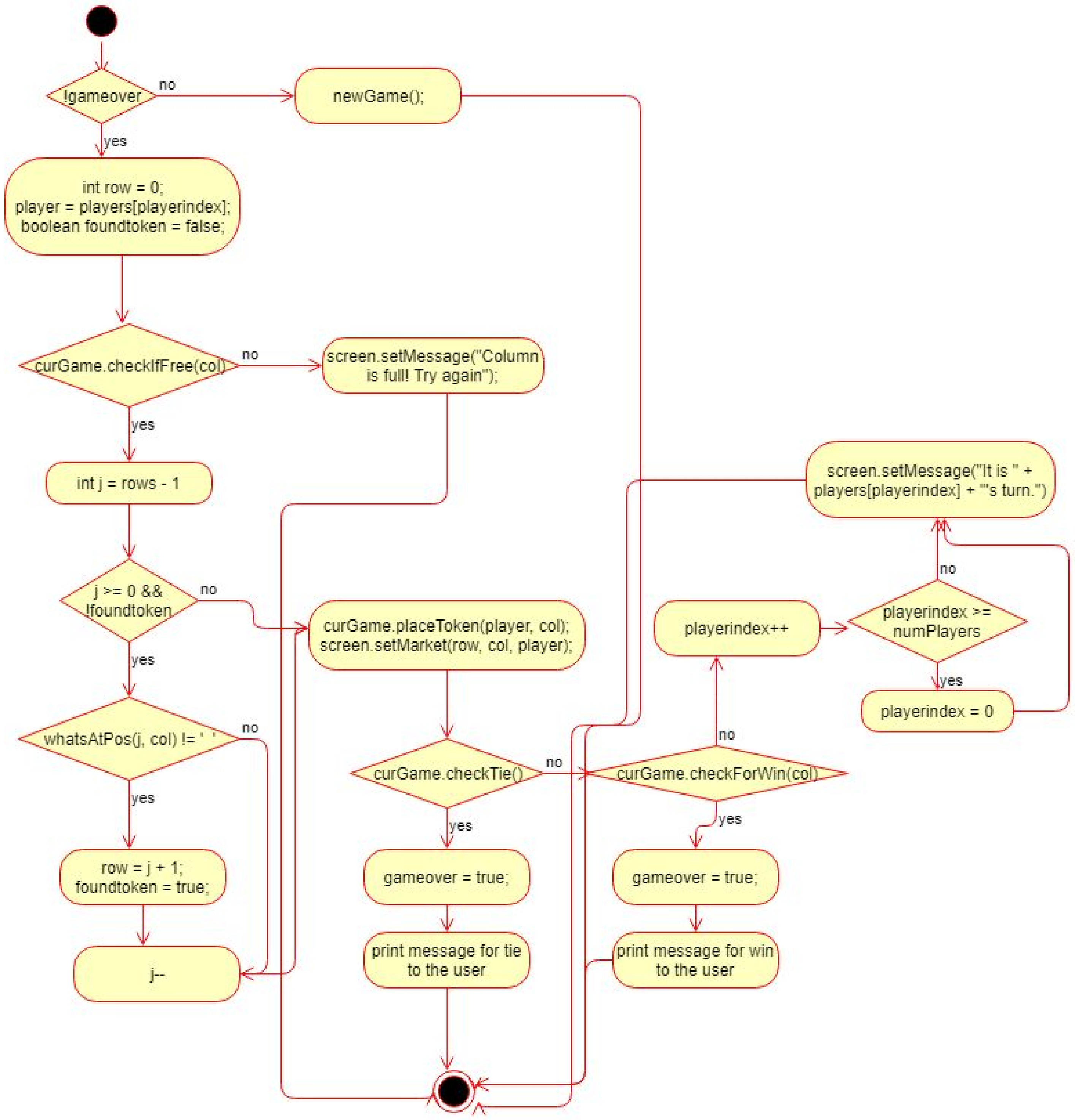


ConnectXController

- IGameboard curGame
- ConnectXView screen
- int MAX_PLAYERS
- char [ ] players
- int numPlayers
- int playerindex
- char player

- ConnectXController(IGameBoard model, ConnectXView view, int np)
- void processButtonClick(int col)
- void newGame()

processButtonClick(int col)


!gameover
no
yes
newGame();
int row = 0;
player = players[playerindex];
boolean foundtoken = false;
curGame.checkIfFree(col)
no
yes
screen.setMessage("Column is full! Try again");
int j = rows - 1
j >= 0 && !foundtoken
no
yes
whatsAtPos(j, col) != ' '
no
yes
row = j + 1;
foundtoken = true;
j--
curGame.placeToken(player, col);
screen.setMarket(row, col, player);
curGame.checkTie()
no
yes
gameover = true;
print message for tie to the user
curGame.checkForWin(col)
no
yes
gameover = true;
print message for win to the user
playerindex++
playerindex >= numPlayers
no
yes
playerindex = 0
screen.setMessage("It is " + players[playerindex] + "'s turn.")