Programming II Assignment 02 Documentation

Adrian Charron

Cégep Heritage College

420-G20-HR: Programming II

Paul Elliott-Magwood
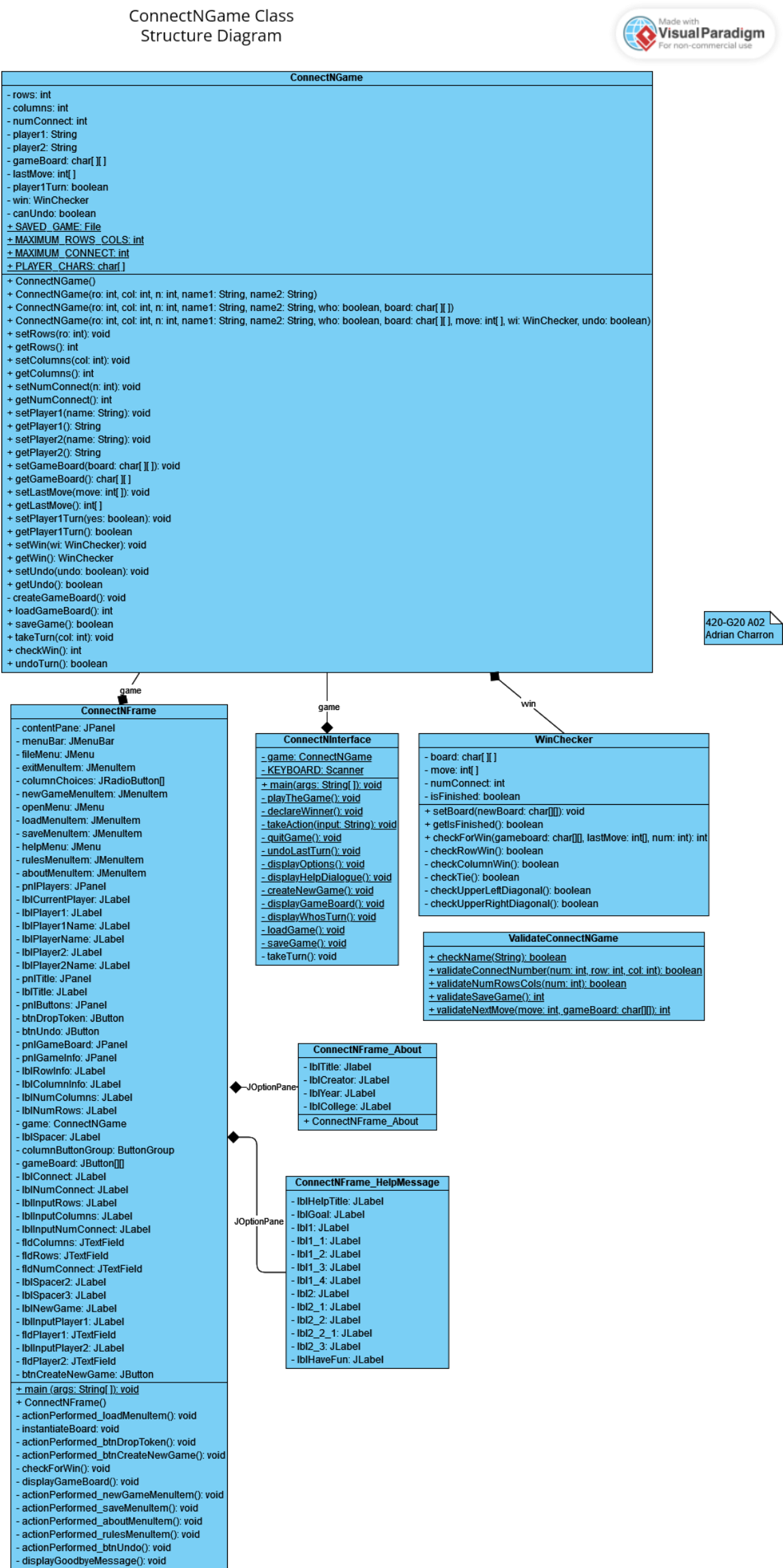
April 26nd, 2024

Table of Contents

Class Structure Diagram

ConnectNGame Class
Structure Diagram

Made with
Visual Paradigm
For non-commercial use

ConnectNGame

- rows: int
- columns: int
- numConnect: int
- player1: String
- player2: String
- gameBoard: char[ ][ ]
- lastMove: int[ ]
- player1Turn: boolean
- win: WinChecker
- canUndo: boolean
+ SAVED_GAME: File
+ MAXIMUM_ROWS_COLS: int
+ MAXIMUM_CONNECT: int
+ PLAYER_CHARS: char[ ]

+ ConnectNGame()
+ ConnectNGame(ro: int, col: int, n: int, name1: String, name2: String)
+ ConnectNGame(ro: int, col: int, n: int, name1: String, name2: String, who: boolean, board: char[ ][ ])
+ ConnectNGame(ro: int, col: int, n: int, name1: String, name2: String, who: boolean, board: char[ ][ ], move: int[ ], wi: WinChecker, undo: boolean)
+ setRows(ro: int): void
+ getRows(): int
+ setColumns(col: int): void
+ getColumns(): int
+ setNumConnect(n: int): void
+ getNumConnect(): int
+ setPlayer1(name: String): void
+ getPlayer1(): String
+ setPlayer2(name: String): void
+ getPlayer2(): String
+ setGameBoard(board: char[ ][ ]): void
+ getGameBoard(): char[ ][ ]
+ setLastMove(move: int[ ]): void
+ getLastMove(): int[ ]
+ setPlayer1Turn(yes: boolean): void
+ getPlayer1Turn(): boolean
+ setWin(wi: WinChecker): void
+ getWin(): WinChecker
+ setUndo(undo: boolean): void
+ getUndo(): boolean
- createGameBoard(): void
+ loadGameBoard(): int
+ saveGame(): boolean
+ takeTurn(col: int): void
+ checkWin(): int
+ undoTurn(): boolean

420-G20 A02
Adrian Charron

game

game

win

ConnectNFrame

- contentPane: JPanel
- menuBar: JMenuBar
- fileMenu: JMenu
- exitMenuItem: JMenuItem
- columnChoices: JRadioButton[]
- newGameMenuItem: JMenuItem
- openMenu: JMenu
- loadMenuItem: JMenuItem
- saveMenuItem: JMenuItem
- helpMenu: JMenu
- rulesMenuItem: JMenuItem
- aboutMenuItem: JMenuItem
- pnlPlayers: JPanel
- lblCurrentPlayer: JLabel
- lblPlayer1: JLabel
- lblPlayer1Name: JLabel
- lblPlayerName: JLabel
- lblPlayer2: JLabel
- lblPlayer2Name: JLabel
- pnlTitle: JPanel
- lblTitle: JLabel
- pnlButtons: JPanel
- btnDropToken: JButton
- btnUndo: JButton
- pnlGameBoard: JPanel
- pnlGameInfo: JPanel
- lblRowInfo: JLabel
- lblColumnInfo: JLabel
- lblNumColumns: JLabel
- lblNumRows: JLabel
- game: ConnectNGame
- lblSpacer: JLabel
- columnButtonGroup: ButtonGroup
- gameBoard: JButton[][]
- lblConnect: JLabel
- lblNumConnect: JLabel
- lblInputRows: JLabel
- lblInputColumns: JLabel
- lblInputNumConnect: JLabel
- fldColumns: JTextField
- fldRows: JTextField
- fldNumConnect: JTextField
- lblSpacer2: JLabel
- lblSpacer3: JLabel
- lblNewGame: JLabel
- lblInputPlayer1: JLabel
- fldPlayer1: JTextField
- lblInputPlayer2: JLabel
- fldPlayer2: JTextField
- btnCreateNewGame: JButton

+ main (args: String[ ]): void
+ ConnectNFrame()
- actionPerformed_loadMenuItem(): void
- instantiateBoard: void
- actionPerformed_btnDropToken(): void
- actionPerformed_btnCreateNewGame(): void
- checkForWin(): void
- displayGameBoard(): void
- actionPerformed_newGameMenuItem(): void
- actionPerformed_saveMenuItem(): void
- actionPerformed_aboutMenuItem(): void
- actionPerformed_rulesMenuItem(): void
- actionPerformed_btnUndo(): void
- displayGoodbyeMessage(): void

ConnectNInterface

- game: ConnectNGame
- KEYBOARD: Scanner
+ main(args: String[ ]): void
- playTheGame(): void
- declareWinner(): void
- takeAction(input: String): void
- quitGame(): void
- undoLastTurn(): void
- displayOptions(): void
- displayHelpDialogue(): void
- createNewGame(): void
- displayGameBoard(): void
- displayWhosTurn(): void
- loadGame(): void
- saveGame(): void
- takeTurn(): void

WinChecker

- board: char[ ][ ]
- move: int[ ]
- numConnect: int
- isFinished: boolean
+ setBoard(newBoard: char[][]): void
+ getIsFinished(): boolean
+ checkForWin(gameboard: char[][], lastMove: int[], num: int): int
- checkRowWin(): boolean
- checkColumnWin(): boolean
- checkTie(): boolean
- checkUpperLeftDiagonal(): boolean
- checkUpperRightDiagonal(): boolean

ValidateConnectNGame

+ checkName(String): boolean
+ validateConnectNumber(num: int, row: int, col: int): boolean
+ validateNumRowsCols(num: int): boolean
+ validateSaveGame(): int
+ validateNextMove(move: int, gameBoard: char[][]): int

JOptionPane

ConnectNFrame_About

- lblTitle: Jlabel
- lblCreator: JLabel
- lblYear: JLabel
- lblCollege: JLabel
+ ConnectNFrame_About

JOptionPane

ConnectNFrame_HelpMessage

- lblHelpTitle: JLabel
- lblGoal: JLabel
- lbl1: JLabel
- lbl1_1: JLabel
- lbl1_2: JLabel
- lbl1_3: JLabel
- lbl1_4: JLabel
- lbl2: JLabel
- lbl2_1: JLabel
- lbl2_2: JLabel
- lbl2_2_1: JLabel
- lbl2_3: JLabel
- lblHaveFun: JLabel

Steps Taken to Test Connect N

Step #:	Actions Taken						
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:	
1. Tried to load game with no saved game			Load		Y	Error message: Cannot load game, file-not-found. Offer new options.	
2. New game: Missing row	4 3			Adrian John		Error message: missing row number Allows user to reenter parameters	
3. New game: Missing column	4 3			Adrian John		Error message: missing column number Allows user to reenter parameters	
4. New game: Missing Connect number	4 4			Adrian John		Error message: missing connect number. Allows user to reenter parameters	
5. New game: Missing player1 name	4 4 3			John		Error message: missing player1 name Allows user to reenter player 1 name.	
6. New game: Missing player2 name	4 4 3			Adrian		Error message: missing player2 name Allows user to reenter player 2 name.	
7. New game: Below minimum rows, Valid columns, Valid N	3 4 3			Adrian John		Error message: invalid number of rows. Allows user to reenter parameters of the game	
8. New game: Above maximum rows, Valid columns, Valid N	13 4 3			Adrian John		Error message: invalid number of rows. Allows user to reenter parameters of the game	

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
9. New game: Valid rows, Below minimum columns Valid N	4 3 3			Adrian John		Error message: invalid number of columns. Allows user to reenter parameters of the game
10. New game: Valid rows, Above maximum columns Valid N	4 13 3			Adrian John		Error message: invalid number of columns. Allows user to reenter parameters of the game
11. New game: Valid rows, Valid columns Below minimum N	4 4 2			Adrian John		Error message: invalid number of columns. Allows user to reenter parameters of the game
12. New game: Maximum rows, Maximum columns Above maximum N	12 12 10			Adrian John		Error message: invalid connect number. Allows user to reenter parameters of the game
13. New game: Valid rows, Valid columns N Above maximum col/row	4 4 6			Adrian John		Error message: invalid connect number. Allows user to reenter parameters of the game

The table header "Actions Taken" spans the columns # rows/# columns/# to win, Column of token dropped, and Save, Quit, Load, Undo?

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					
14. New game: Valid rows, Valid Columns, N == col/row	4 4 4			Adrian John		Game successfully created with: 4 4 4 Adrian John E~E~E~E E~E~E~E E~E~E~E E~E~E~E
15. Player1 turn Column below minimum	4 4 4	0 (index -1)		Adrian John		Error Message: Invalid column number. Please reenter. Still player 1's turn.
16. Player1 turn Column over maximum	4 4 4	5 (index 4)		Adrian John		Error Message: Invalid column number. Please reenter. Still player 1's turn.
17. Continue game: Player1 turn Save Game	4 4 4		Save	Adrian John		Game was successfully saved: 4 4 4 Adrian John 1 E~E~E~E E~E~E~E E~E~E~E E~E~E~E

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					
18. Quit game			Quit			Quit successfully. Thanks for playing!
19. Load game, file corrupted.			Load			Error message: File is corrupted. Game cannot be loaded.
20. New game:	4 4 4			Adrian John		Game successfully created: 4 4 4 Adrian John E~E~E~E E~E~E~E E~E~E~E E~E~E~E
21. Player 1 tries to undo			Undo	Adrian John		Error Message: Cannot undo a move after the beginning of a game. Still player 1's turn.
22. Player1 turn Plays and confirms	4 4 4	Column 2 (Index 1)		Adrian John		Play confirmed, next player's turn E~E~E~E E~E~E~E E~E~E~E E~<mark>Y</mark>~E~E
23. Continue game: Player2 turn	4 4 4	Column 2 (Index 1)		Adrian John		Play confirmed, next player's turn E~E~E~E E~E~E~E E~<mark>R</mark>~E~E E~Y~E~E

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					

24. Player 1's turn
    Player 2 undoes previous turn
    Return to player 2's turn
			Undo	Adrian
				John
					Y	Undo confirmed, board updated:
						E~E~E~E
						E~E~E~E
						E~**E**~E~E
						E~Y~E~E
						Return to player 2's turn.

25. Continue game:
    Player2 turn
    Plays and confirms
	4
	4
	4
		Column 1 (Index 0)		Adrian
				John
						Play confirmed, next player's turn
						E~E~E~E
						E~E~E~E
						E~E~E~E
						R~Y~E~E

26. Player1 wins (col) Game ends		Column 2 (Index 1)		Adrian John	Y	Play confirmed, Winner: Adrian (Player 1) E~**Y**~E~E R~Y~E~E R~Y~E~E R~Y~E~E New game, load game, or quit?
27. Player 1 tries to undo			Undo	Adrian John	Y	Error message: Cannot undo once a game is over.
28. Quit game			Quit		Y	Thank you for playing. Program ends
29. Load game: Player 1's turn	4 4 4		Load	Adrian John	Y	Game loaded successfully: Player 1's turn E~E~E~E E~E~E~E E~E~E~E E~E~E~E

Step #:	Actions Taken					
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
30. Player1 wins (row) Game ends		Column 4 (index 3)		Adrian John	Y	Winner: Adrian (Player 1) E~E~E~E E~E~E~E R~R~R~E Y~Y~Y~<mark>Y</mark> New game, load game, or quit?
31. Player 1 tries to undo			Undo	Adrian John	Y	Error message: Cannot undo once a game is over.
32. Quit game			Quit		Y	Thank you for playing. Program ends
33. Load game: Player 1's turn	4 4 4		Load	Adrian John	Y	Game loaded successfully: Player 1's turn E~E~E~E E~E~E~E E~E~E~E E~E~E~E
34. Invalid move: Player 1's turn Column full		Column 2 (index 1)		Adrian John	Y	Error message: Please enter a valid column Y~R~E~E R~Y~E~E Y~R~E~E R~Y~E~E Still player 1's turn
35. Player 2's turn, Game is a tie: All fields filled		Column 3 (index 2)		Adrian John	Y	Game is a draw. Y~R~<mark>R</mark>~Y R~Y~Y~R Y~R~R~Y R~Y~Y~R New game, load game, or quit?

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					
36. Player 2 tries to undo			Undo	Adrian John	Y	Error message: Cannot undo once a game is over.
37. New game: 12 rows, 12 col, connect 8	12 12 8			Adrian John		Game successfully created with: 4 4 4 Adrian John E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E

Step #:	Actions Taken					Output:
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	
38. Player 1's turn: plays and confirms		Column 1 (index 0)		Adrian John	Y	Move Successful: E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E <mark>Y</mark>~E~E~E~E~E~E~E~E~E~E~E

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					
39. Save game, confirmed: Player 2's turn	12 12 8		Save	Adrian John	Y	Game successfully saved: 4 4 4 Adrian John 2 E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E Y~E~E~E~E~E~E~E~E~E~E~E
40. Quit game Confirmed			Quit	Adrian John	Y	Game closes, Thank you for playing

Step #:	Actions Taken					Output:
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	
41. Load game from save: Player 2's turn	12 12 8		Load	Adrian John	Y	Game loaded successfully: 4 4 4 Adrian John 2 E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E Y~E~E~E~E~E~E~E~E~E~E~E
42. Player 1 tries to undo			Undo	Adrian John		Error message: Cannot undo after a game has been loaded.

Step #:	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:
	Actions Taken					
43. Player2 wins (Upper left to lower right) Game ends		Column 12 (index 11)		Adrian John	Y	Winner John! (Player 2) E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~R ~E~E~E~E~E~E~E E~E~E~E~Y~R~E~E~E~E~E~E E~E~E~E~R~Y~R~E~E~E~E~E E~E~E~E~Y~R~Y~R~E~E~E~E E~E~E~E~R~Y~R~Y~R~E~E~E E~E~E~E~Y~R~Y~R~Y~R~E~E E~E~E~E~R~Y~R~Y~R~Y~R~E Y~Y~Y~Y~Y~R~Y~R~Y~R~Y~<mark>R</mark> New game, load game, quit?
44. Player 2 tries to undo			Undo	Adrian John	Y	Error message: Cannot undo once a game is over.

| Step #: | Actions Taken | | | | | |
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	Output:

45. New game:
    12 rows,
    12 col,
    connect 8
	12
	12
	8
				Adrian
				John
						Game successfully created with:
						4
						4
						4
						Adrian
						John
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E
						E~E~E~E~E~E~E~E~E~E~E~E

| Step #: | Actions Taken | | | | | Output: |
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	
46. Player2 wins (upper right to lower left) Game ends		Column 1 (index 0)		Adrian John	Y	John Wins! (Player 2) E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~R ~E~E~E~E E~E~E~E~E~E~R~Y~E~E~E~E E~E~E~E~E~R~Y~R~E~E~E~E E~E~E~E~R~Y~R~Y~E~E~E~E E~E~E~R~Y~R~Y~R~E~E~E~E E~E~R~Y~R~Y~R~Y~E~E~E~E E~R~Y~R~Y~R~Y~R~E~E~E~E R~Y~R~Y~R~Y~R~Y~Y~Y~Y~Y New game, load game, quit?
47. Player 2 tries to undo			Undo	Adrian John	Y	Error message: Cannot undo once a game is over.

Step #:	Actions Taken					Output:
	# rows: **# columns:** **# to win:**	**Column of token dropped:**	**Save, Quit, Load, Undo?**	**Player1 name:** **Player2 name:**	**Confirm**	
48. New game:	12 12 8			Adrian John		New game successfully created: 4 4 4 Adrian John E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E

| Step #: | Actions Taken | | | | | Output: |
	# rows: # columns: # to win:	Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	
49. Player 1 wins (Upper left to lower right middle of the board) Game ends		Column 7 (index 6)		Adrian 30 John	Y	Adrian Wins! (Player 1) E~E~E~E~E~E~E~E~E~E~E~E E~E~E~E~E~E~E~E~E~E~E~E E~E~Y~E~E~E~E~E~E~E~E E~E~R~Y~E~E~E~E~E~E~E E~E~Y~R~Y~E~E~E~E~E~E E~E~R~Y~R~Y~E~E~E~E~E E~E~Y~R~Y~R~<mark>Y</mark>~E~E~E~E~E E~E~R~Y~R~Y~R~Y~E~E~E~E E~E~Y~R~Y~R~Y~R~Y~E~E~E E~E~R~Y~R~Y~R~Y~R~Y~E~E E~E~Y~R~Y~R~Y~R~Y~Y~R~R R~R~R~Y~R~Y~R~Y~R~R~Y~R New game, load game, quit?
50. New game:	5 7 3			Adrian John		Game successfully created: 5 6 3 Adrian John E~E~E~E~E~E~E E~E~E~E~E~E~E E~E~E~E~E~E~E E~E~E~E~E~E~E E~E~E~E~E~E~E

Step #:	# rows: # columns: # to win:	Actions Taken				Output:
		Column of token dropped:	Save, Quit, Load, Undo?	Player1 name: Player2 name:	Confirm	
51. Player 2 wins (Upper right to lower left, middle of the board) Game ends.		Column 3 (index 2)		Adrian John	Y	John Wins! (Player 2) E~E~E~E~E~E~E E~E~E~E~R~E~E E~E~E~<mark>R</mark>~Y~E~E Y~E~R~Y~R~E~E Y~Y~R~R~Y~E~E
52. Quit game after a win			Quit		Y	Game closes, Thank you for playing