CheckerBoard(void) – testCheckBoard_Initialize

Input: N/A

State: N/A

Output: [Board initialized]

State:

x	*	x	*	x	*	x	*
*	X	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MinRow_MinCol_Player_1

Input: pos = (0, 0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘x’

State: [Board state is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MaxRow_MaxCol

Input: pos = (7, 7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: “o”

State: [Board state is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_Empty_Space

Input: pos = (3, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘ ‘

State: [Board state is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_Black_Tile

Input: pos = (0, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘*’

State: [Board state is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_After_Turn_1_Player_1

Input: pos = (3, 3)

State: [Player 1 moved from (2,2) to (3,3)]

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*		*	x	*	x	*
*		*	x	*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘x’

State: [Board state is unchanged]

placePiece(BoardPosition pos, char player) – testPlacePiece_Empty_Tile _Player_1

Input: pos = (3, 1), player = ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*	x	*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Occupied_Tile_Empty

Input: pos = (5, 1), player = ‘ ’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*		*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Empty_TIle_Player_2

Input: pos = (4, 6), player = ‘o’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*	o	*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_After_Turn_1_Player_2

Input: pos = (2, 4), player = ‘o’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*	x	*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	o	*	x	*
*		*		*	x	*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_After_Turn_2_Player_1

Input: pos = (5, 7), player = ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*	x	*	
	*		*		*	o	*
*	o	*	o	*	o	*	
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*	x	*	
	*		*		*	o	*
*	o	*	o	*	o	*	x
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

getPieceCounts(void) – getPieceCounts_Initial_Board

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: {‘x’=12, ‘o’=12}

State: [Board state is unchanged]

getViableDirections(void) – testGetViableDirections

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: {‘x’=[SE, SW], ‘X’=[SE, SW, NE, NW], ‘o’=[NE, NW], ‘O’=[SE, SW, NE, NW]}

State: [Board state is unchanged]

checkPlayerWin(Character player) – testCheckPlayerWin_False_Player_1

Input: player = ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: false

State: [Board state is unchanged]

checkPlayerWin(Character player) – testCheckPlayerWin_True_Player_1

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

Output: true

State: [Board state is unchanged]

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Back_Row_Player_1

Input: posOfPlayer = (7, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	x	*	o	*	o	*	o

Output:

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	X	*	o	*	o	*	o

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Middle_Board_Player_1

Input: posOfPlayer = (3, 3)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*	x	*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*	X	*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Back_Row_Player_2

Input: posOfPlayer = (0, 2)

State:

x	*	o	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*		*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

State:

x	*	O	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*		*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMoviePiece_Player_1_SE

Input: startingPos = (2, 0), dir = SE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (3, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
	*	x	*	x	*	x	*
*	x	*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece _Player_2_NE

Input: startingPos = (5, 5), dir = NE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (4, 6)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*	o	*
*	o	*	o	*		*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_Crowned_Piece_SW

Input: startingPos = (4, 4), dir = SW

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*		*	x	*	x	*
*	x	*		*		*	
	*		*	O	*		*
*	o	*		*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (5, 3)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	O	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Player_1_Jump_Player_2

Input: startingPos = (3, 3), dir = SW

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*		*	x	*	x	*
*		*	x	*		*	
	*	o	*		*		*
*		*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (5, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*		*	x	*
*		*		*		*	
	*		*		*		*
*	x	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Player_2_Jump_Player_1

Input: startingPos = (5, 3), dir = NW

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*		*	x	*	x	*
*		*		*		*	
	*	x	*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (3, 1)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*		*	x	*	x	*
*	o	*		*		*	
	*		*		*		*
*	o	*		*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Crowned_Piece_Jump_Backwards

Input: startingPos = (5, 5), dir = NE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*		*
*		*		*		*	
	*		*		*	o	*
*	o	*	o	*	X	*	
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: (3, 7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*		*
*		*		*		*	X
	*		*		*		*
*	o	*	o	*		*	
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_All_Empty

Input: startingPos = (4, 2)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*	o	*	o	*		*
*		*		*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: {SE=’ ‘, SW=’ ‘, NE= ‘ ‘, NW=’ ‘}

State: [Board state is unchanged]

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_All_Occupied

Input: startingPos = (6, 2)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: {SE=’o‘, SW=’o‘, NE= ‘o‘, NW=’o‘}

State: [Board state is unchanged]

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_Board_Edge

Input: startingPos = (2, 0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: {SE=’ ‘, SW= “Off The Board”, NE= ‘x‘, NW= “Off The Board”}

State: [Board state is unchanged]