## CheckerBoard(void) – [Test\_Constructor]

Input: N/A	Outp	out: N	/A					
State: N/A		eCour leDire					= NW	٧,
	X	*	Х	*	х	*	х	*
	*	Х	*	Х	*	Х	*	Х
	x	*	Х	*	Х	*	Х	*
	*		*		*		*	
		*		*		*		*
	*	0	*	0	*	0	*	0
	0	*	0	*	0	*	0	*
	*	0	*	0	*	0	*	О

#### $whats At Pos(Board Position\ pos) - test Whats At Pos\_MaxRow\_MinCol\_Black Tile$

Inpu	ıt: (7,0	))						Outp		*'					
State	e:							X	-   *	X	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	x	*	X	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0		0		<u> </u>		10		10

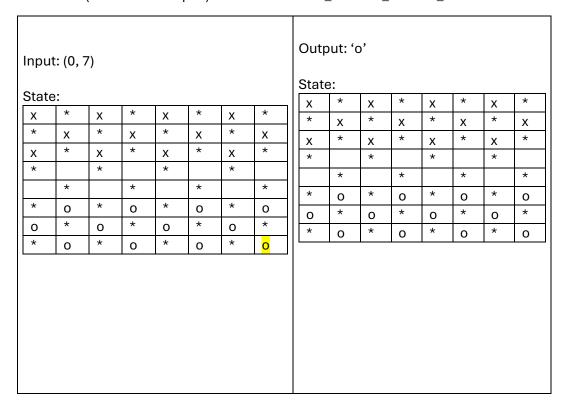
 $whats At Pos (Board Position\ pos) - test Whats At Pos\_MinRow\_MinCol\_White Tile$ 

State: x *	.							e:						
X *	L.		_		1		х	*	Х	*	Х	*	Х	*
	^	*	Х	*	Х	*	*	х	*	х	*	Х	*	х
* X	<b>*</b>	Х	*	Х	*	Χ	X	*	Х	*	Х	*	Х	*
x *	k X	*	Х	*	Х	*	*		*		*		*	
*	*		*		*			*		*		*		*
*	<b>k</b>	*		*		*	*	0	*	0	*	0	*	0
* 0	o *	0	*	0	*	0	0	*	0	*	0	*	0	*
0 *	* O	*	0	*	0	*	*	0	*	0	*	0	*	0
* 0	o *	0	*	0	*	0		U		U		U		U

 $whats At Pos(Board Position\ pos) - test Whats At Pos\_Max Row\_Max Col\_Black Tile$ 

Inpu	it: (7,	7)						Out <sub>l</sub> Stat		*'					
State	e:							X	*	X	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	x	*	х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	<u>^</u>		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*		*	1	*		*	
*	О	*	О	*	0	*	0		0 *	-	0 *	1	0 *	1	0 *
0	*	0	*	О	*	О	*	0 *	-	0 *	1	0 *		0 *	
*	0	*	0	*	0	*	0		0		0	1	0		0

#### $whats At Pos(Board Position\ pos) - test Whats At Pos\_MinRow\_Max Col\_White Tile$



### $whats At Pos (Board Position\ pos) - test Whats At Pos\_Mid Row\_Mid Col\_White Tile$

npu	t: (3,	3)						Outp		•					
State	٦.							State		1	Τ.		Τ.		T .
X	*	Х	*	Х	*	Х	*	X *	*	X *	*	X *	*	X *	*
*	х	*	Х	*	х	*	х		Х		Х		Х	1	X
X	*	Х	*	Х	*	Х	*	X	*	X	*	X	*	X	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0
	*		*		*		*	0	*	0	*	0	*	0	*
0 *	0	0 *	0	0 *	0	0 *	0	*	0	*	0	*	0	*	0
	1	1	1	1			ı								

placePiece(BoardPosition pos, char player) – [test\_placePiece\_pos07\_x]

Inpu	ıt: [0,	7] x						Outp	out: n.	/a					
Stat	e:							State	e:						
Doo	al							Boar	d =						
Boai	ru =   *		*		*	1		X	*	Х	*	Х	*	Х	×
Х		Х		Х	+	Х	*	*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0				10		10		U

placePiece(BoardPosition pos, char player) – [test\_placePiece\_pos00\_o]

Input	e:	)] o						Outp State Boar		/a					
Boar	u –   *		*		*	Τ,,	*	O	*	Х	*	Х	*	Х	*
X *		X *	-	X *	-	X *		*	Х	*	Х	*	Х	*	Х
	X		X		X		X	Х	*	Х	*	Х	*	Х	*
Χ	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	О	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	О	*	0	*	0	*	0
*	0	*	0	*	0	*	0		10		10		10		0

 $place Piece (Board Position pos, char player) - [test\_place Piece\_pos 70\_x]$ 

npu	t: [7,0	)] x						State	out: n	/a					
State	e:							State	<b>5.</b>						
_								Boar	rd =						
Boar	_	1	1.	1	1.	1		Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	х
*	Х	*	Х	*	Х	*	Х	X	*	Х	*	Х	*	Х	*
Χ	*	Х	*	Х	*	Х	*	*		*	+	*	+	*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	+	*	-	*	_	*	+
*	0	*	0	*	0	*	0		0 *		0 *	-	0 *	-	0 *
0	*	0	*	0	*	0	*	0		0		0		0	
*	0	*	0	*	0	*	0	×	0	*	0	*	0	*	0
	U		U		U		10								

 $place Piece (Board Position\ pos,\ char\ player) - [test\_place Piece\_pos 77\_x]$ 

Input		7] x						State		/a					
Boar	d =							Boar	,	1	_		_	1	
l <del></del>	u –   *	T .,	*	Τ.,	*	Τ.,	*	Х	*	Х	*	Х	*	Х	*
X		X		X		X		*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	Х	*	Х	*	Х	*	Х	*
х	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*		*		*		*	<u> </u>
*	0	*	0	*	0	*	0		0		0	^	0		0
	*		*	+	*	-	*	0	*	0	*	0	*	0	*
0	^	0		0		0		*	0	*	0	*	0	*	<mark>x</mark>
*	0	*	0	*	0	*	O		1	1	1	1	1	1	

placePiece(BoardPosition pos, char player) – [test\_placePiece\_pos44\_o]

Input State Board	:	4] o						State Boar		X	*	х	*	Х	*
Х	*	Х	*	X	*	X	*	*	X *	*	X *	*	X *	*	X *
*	Х	*	Х	*	х	*	Х	X	*	X	_ ^	X	^	X	^
	*	х	*	· ·	*	· ·	*	*		*		*		*	
X		1	-	X	-	X *			*		*	O	*		*
*		*		*		*		*	0	*	0	*	0	*	0
	*		*		*		*	0	*	0	*	0	*	0	*
*	О	*	0	*	0	*	0	*	0	*	0	*	0	*	
0	*	0	*	0	*	0	*	L	U	1	U	1	U		0
*	0	*	0	*	0	*	О								

 $getPieceCounts(void) - [Test\_getPieceCounts\_12]$ 

Input	t: N/A	<b>\</b>						Outp	out: 1	2, 12					
viabl boar	eCou eDire d =	ection	is = x		, SE; (	ı	V, NE	viabl NE	eCou leDire	nt = x ection				o = NV	۷,
Х	*	Х	*	Х	*	Х	*	boar	Ť	1			1.	1	1 .
*	Χ	*	Х	*	Х	*	Х	Х	*	Х	*	Χ	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	Х
*		*		*		*		Х	*	Х	*	Х	*	Х	*
	*		*		*		*	*		*		*		*	
*	0	*	0	*	0	*	0		*		*		*		*
0	*	О	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
	•	•	•	•	•			*	0	*	0	*	0	*	0

 $\verb"getViableDirections" (void) - [Test\_getViableDirections\_start]$ 

Input	t: N/a							Out	put: x	= SW	SE;	o = NV	V, NE		
	eCou eDire			, o = 1 = SW		o = NV	V, NE	1 -	e: eCou leDire					= NV	۷,
Х	*	Х	*	Х	*	Х	*	boa	rd =						
*	Х	*	Х	*	Х	*	Х	Х	*	Χ	*	Х	*	Х	*
Χ	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	Х
*		*		*		*		Х	*	Х	*	Х	*	Х	*
	*		*		*		*	*		*		*		*	
*	0	*	0	*	0	*	0		*		*		*		*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
		•	•	•	•	•		*	0	*	0	*	0	*	0

 $check Player Win (Character\ player) - [test\_check Player Win\_player XWin]$ 

Input	t: acter	play	er = ";	x"				Output: Returns true
			-	-		/ all "	o"	State: No change.
х	*	Х	*	Х	*	Х	*	
*	Х	*	Х	*	Х	*	Х	
Х	*	Х	*	Х	*	Х	*	
*		*		*		*		
	*		*		*		*	
*		*		*		*		
	*		*		*		*	
*		*		*		*		

 $check Player Win (Character\ player) - [test\_check Player Win\_player X Losing]$ 

Inpu <sup>-</sup> Char	t: acter	playe	er = ">	<b>(</b> "				Output: Returns false
State Boar "x" p	d con	tains	"o" p	ieces	and	a sing	ular	State: No change.
	*		*		*		*	
*		*		*		*		
	*		*		*		*	
*		*		*		*		
	*		*		*		*	
*	0	*	0	*	0	*	0	
0	*	0	*	0	*	0	*	
*	0	*	0	*	0	*	0	

 $crown Piece (Board Position\ pos Of Player) - [test\_crown Piece\_player Crown]$ 

Input Boar		ition(	7, 1)					Oi Na	•	ut:						
State		werc	ase ("	'x") at	t posi	tion (7	7, 1)	Po	ре	e: ion s ercase o refl	e cha	racte	er (so	"x" k	eco	
х	*	Х	*	Х	*	Х	*	Х		*	Х	*	Х	*	Х	*
*	Х	*	Х	*	Х	*	Х	*		Х	*	Х	*	Х	*	Х
Х	*	Х	*	Х	*	Х	*	Х	,	*	Х	*	Х	*	Х	*
*		*		*		*		*			*		*		*	
	*		*		*		*			*		*		*		*
*	0	*	0	*	0	*	0	*		0	*	0	*	0	*	0
0	*	0	*	0	*	0	*	C	)	*	0	*	0	*	0	*
*	0	*	0	*	0	*	0	*	•	0	*	0	*	0	*	0

crownPiece(BoardPosition posOfPlayer) - [test\_crownPiece\_noCrown]

Input Boar		ition(	3, 3)					Outp N/A	ut:						
State		werc	ase at	t posi	tion (	3, 3)		State No c	e: hange	э.					
								Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0								

crownPiece(BoardPosition posOfPlayer) - [test\_crownPiece\_edge]

Input	t: dPosi	ition (	0, 0)					Outp N/A	out:						
State		werc	ase a	t posi	tion ((	0, 0)		uppe	tion s	e cha	racte	er (so	"o" l	эесо	
х	*	Х	*	Х	*	Х	*	X	*	Х	*	Х	*	Х	*
*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х
Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	0	*	0	*	0	*	0	*	0	*	0	*	0	*	О
0	*	0	*	0	*	0	*	0	*	0	*	0	*	0	*
*	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0

 $crown Piece (Board Position\ pos Of Player) - [test\_crown Piece\_crown Twice]$ 

Inpu Boar	t: dPosi	tion (	0, 0)					Output: N/A
State Piec	e: e is up	perc	ase a	t posi	tion (	0, 0)		State: No change.
Χ	*	Х	*	Х	*	Х	*	
*	Х	*	Х	*	Х	*	Х	
Х	*	Х	*	Х	*	Х	*	
*		*		*		*		
	*		*		*		*	
*	0	*	0	*	0	*	0	
0	*	0	*	0	*	0	*	
*	0	*	0	*	0	*	0	

 $move Piece (Board Position\ starting Pos,\ Direction Enum\ dir) - [test\_move Piece\_pos 00]$ 

State		0] SE						Outp State Boar	e:	,1]					
Boar		1	1		T	1	T 1		*	Х	*	Х	*	Х	*
X	*	Х	*	Х	*	Х	*	*	X	*	Х	*	Х	*	Х
*		*	Х	*	Х	*	Х	Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*		*		*	
*	Х	*		*		*			*		*		*		*
	*		*		*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	О	0	*	0	*	0	*	0	*
0	*	0	*	0	*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0		10		10		10		U

 $move Piece (Board Position\ starting Pos,\ Direction Enum\ dir)-[test\_move Piece\_pos 77]$ 

Inpu	t: [7,	7] NW	/					Outp	out:[6	,6]					
State	э:							State	э:						
Boar	d =							Boai	d =						
Х	*	Х	*	Х	*	Х	*	Х	*	Χ	*	Х	*	Х	*
*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х
Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*	Х	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0
0	*	0	*	О	*		*	0	*	0	*	0	*	O	*
*	0	*	0	*	0	*	O	*	0	*	0	*	0	*	

 $move Piece (Board Position\ starting Pos,\ Direction Enum\ dir)-[test\_move Piece\_pos 53]$ 

Inpu	t: [5, (	3] NE						Outp	ut: [4	l,4]					
State	e:							State	):						
Boar	d =							Boar	d =						
Х	*	Х	*	Х	*	Х	*	Boar	<b>.</b>						
*	Х	*	Х	*	Х	*	Х	х	*	х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	х	*	х	*	Х	*	X
*		*		*		*		Х	*	х	*	Х	*	Х	*
	*		*		*		*	*		*		*		*	
*	0	*	o	*	0	*	0		*		*	0	*		*
0	*	0	*	0	*	0	*	*	0	*		*	0	*	0
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
								*	0	*	0	*	0	*	0
									1 -	_1	1 -	1	1 -	1	

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – [Test\_jumpPieceFrom\_13SW]

State		] SW						Outr State Boar	e:	2,2]					
Boar			Τ.		Τ.		Τ.	Х	*	Х	*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	Х	*		*	Х	*	Х
*	Х	*	×	*	Х	*	Χ	х	*		*	Х	*	Х	*
Х	*	Х	*	Х	*	Х	*	*	X	*		*		*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	0
*	0	*	0	*	О	*	0	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0						10		U

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – [Test\_jumpPieceFrom\_60NE]

Inpu	t:[6,0	] NE						Outp	out: [	4,2]					
State								State Boar							
Boar	a =   *	1	*	1	*		*	Х	*	Χ	*	Х	*	Х	*
Х		Х		Х		X		*	Х	*	Х	*	Х	*	Х
*	Х	*	Х	*	Х	*	Х	Х	*	Х	*	Х	*	Х	*
Χ	*	Х	*	Х	*	Х	*	*		*		*		*	
*		*		*		*			*	O	*		*		*
	*		*		*		*	*		*	0	*	0	*	0
*	0	*	0	*	0	*	0		*	0	*	0	*	0	*
<mark>0</mark>	*	0	*	0	*	0	*	*	0	*	0	*	0	*	0
*	0	*	0	*	0	*	0	L			U				10

 $jump Piece (Board Position\ starting Pos,\ Direction Enum\ dir)-[Test\_jump Piece From\_13SW]$ 

npu	ıt: [1,	3] SE						Outp	out: [	3,5]					
2+2+								State	э:						
State Boar								Boar	1	1	,	1	,	,	
X	w =	Х	*	X	*	Х	*	Х	*	Х	*	Х	*	Х	7
*	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	*		*		*		*	Χ	*		*	Х	*	>
	X *	-	X *		X *	-	X *	Х	*	Х	*		*	Х	*
Х	*	Х	^	Х	*	Х		*		*		*	X	*	
*		*		*		*			*		*		*		*
	*		*		*		*	*	0	*	0	*	0	*	(
*	0	*	0	*	0	*	0	0	*	0	*	0	*	0	*
0	*	0	*	0	*	0	*	*	0	*	0	*	0	*	(
*	0	*	0	*	0	*	0	L	U		U				,

scanSurroundingPositions(BoardPosition startingPos) – test\_scanSurroundingPositions\_MidRow\_MidCol\_WhiteTile

Input: (3,3)  State:    X									ction ction	Enur Enur Enur Enur	n.NE n.SW	= 'x' ' = ' '			
Х	*	Х	*	Х	*	Х	*	State:							
*		*		*		*		Х	*	Х	*	Х	*	Х	*
	*		*		*		*	*	Х	*	Х	*	Х	*	Х
*	0	*	0	*	0	*	0	Х	*	Х	*	Х	*	Х	*
0	*	0	*	0	*	0	*	*		*		*		*	
*	0	*	0	*	0	*	0		*		*		*		*
								*	0	*	0	*	0	*	0
								0	*	0	*	0	*	0	*
								*	0	*	0	*	0	*	0

 $scan Surrounding Positions (Board Position starting Pos) - \\test\_scan Surrounding Positions\_minRow\_MaxCol\_Black Tile$ 

Input		7)					Dire	ction ction	Enun	n.NE	= N/A	A			
X	*	Х	*	Х	*	Х	DirectionEnum.SW = '*'								
*	х	*	Х	*	DirectionEnum.NW = N/A										
Х	*	х	*	Х	*	Х	*	-							
*		*		*		*		State	e:						
	*		*		*		*	X	*	х	*	Х	*	Х	*
*	0	*	0	*	0	*	0	*	Х	*	Х	*	Х	*	Х
0	*	0	*	0	*	0	*	Х	*	Х	*	Х	*	Х	*
*	0	*	0	*	0	*	0	*		*		*		*	
									*		*		*		*
								*	0	*	0	*	0	*	0
								0	*	0	*	0	*	0	*
								*	0	*	0	*	0	*	0

 $scan Surrounding Positions (Board Position starting Pos) - \\test\_scan Surrounding Positions\_MaxRow\_Second Col\_White Tile$ 

npu <sup>:</sup> State	t: (7, <sup>·</sup>	1)					Dire	ction ction	Enun	n.NE					
Х	*	Х	*	Х	*	Х	DirectionEnum.SW = N/A DirectionEnum.NW = 'o'								
*	Х	*	Х	*	Х	*	Х								
Χ	*	Х	*	Х	*	Х	*	State:							
*		*		*		*		Х	*	Х	*	Х	*	Х	*
	*		*		*		*	*	Х	*	Х	*	Х	*	Х
*	0	*	0	*	0	*	0	Х	*	Х	*	Х	*	Х	*
0	*	0	*	0	*	0	*	*		*		*		*	
*	0	*	0	*	0	*	0		*		*		*		*
								*	0	*	0	*	0	*	0
								0	*	0	*	0	*	0	*
								*	0	*	0	*	0	*	0

# ${\tt getDirection(DirectionEnum\,dir)-[test\_getDirection\_southeastSE]}$

Input:	Output:
DirectionEnum.SE	Returns BoardPosition(4, 4)
	(Position has moved 1 row down, 1
State:	column right)
Piece is at position (3, 3)	
	State:
	No change.

What tests did each team member write? Just tell me the names of the functions (unless for some reason multiple team members wrote functions for the same method. In that case, tell me which tests specifically by giving me the test names)

Elise James	whatsAtPos(BoardPosition) scanSurroundingPositions(BoardPosition)
Jada Hall	placePiece(BoardPosition, char) movePiece(BoardPosition, DirectionEnum)
Molly McKenzie	CheckerBoard(int) getPieceCounts(void) getViableDirections(void) jumpPiece(BoardPosition, Direction Enum)
Sarah Mendoza	getRowNum(void) getColNum(void) checkPlayerWin(Character) crownPiece(BoardPosition) getDirection(DirectionEnum)