/* Tahimic & Griffin(S20)
MPCCPROG3_MCO2_Animals&Methods_Test Cases*/

ANIMALS
Animal : mouse

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	can't cross river	BLUE Team (6,2) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can't move to own den	BLUE Team (3,1) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (3,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
3	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (6,6) to the Left	New Row = 6 New Column = 5	New Row = 6 New Column = 5	Pass
4	can go to it's own trap	BLUE Team (3,2) to the Left	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,5) to the Right	New Row = 3 New Column = 6	New Row = 3 New Column = 6	Pass
5	can't go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can't land on position occupied by teammate	BLUE Team (6,1) to the Left BLUE Lion : (6,0)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,7) Upward RED leopard: (1,7)	[INVALID MOVE]	[INVALID MOVE]	Pass

7	will be captured by enemy with higher rank	BLUE Team (3,2) RED Wolf : (3,3) to the Right	mouse will be captured	mouse was captured	Pass
		RED Team (1,5) BLUE Tiger : (1,2) to the Right	mouse will be captured	mouse was captured	Pass
8	will capture enemy with equal rank	BLUE Team (2,4) Upward RED Mouse : (1,4)	New Row = 1 New Column = 4	New Row = 1 New Column = 4	Pass
		RED Team (1,4) Downward BLUE Mouse : (2,4)	New Row = 2 New Column = 4	New Row = 2 New Column = 4	Pass
9	will be captured by enemy with equal rank	BLUE Team (2,4) RED Mouse : (1,4) Downward	mouse will be captured	mouse will be captured	Pass
		RED Team (1,4) BLUE Mouse : (2,4) Upward	mouse will be captured	mouse will be captured	Pass
10	will be trapped on enemy's trap and captured by enemy's elephant	BLUE Team (3,7) RED Elephant : (4,7) Upward	mouse will be captured	mouse was captured	Pass
		RED Team (2,0) BLUE Elephant : (2,1) to the Left	mouse will be captured	mouse will be captured	Pass
11	will capture elephant	BLUE Team (3,5) to the Right RED Elephant : (3,6)	New Row = 3 New Column = 6	New Row = 3 New Column = 6	Pass
		RED Team (0,5) to the Left RED Elephant : (0,4)	New Row = 0 New Column = 4	New Row = 0 New Column = 4	Pass
12	can move to river	BLUE Team (2,2) to the Right	New Row = 2 New Column = 3	New Row = 2 New Column = 3	Pass
		RED Team	New Row = 5	New Row = 5	Pass

		(6,3) Upward	New Column = 3	New Column = 3	
13	can capture enemy mouse in river as well	BLUE Team (2,4) Upward RED Mouse : (1,4)	New Row = 1 New Column = 4	New Row = 1 New Column = 4	Pass
		RED Team (1,4) Downward BLUE Mouse : (2,4)	New Row = 2 New Column = 4	New Row = 2 New Column = 4	Pass
14	can't capture enemy mouse while in river	BLUE Team (2,5) Downward RED Mouse : (3,5)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,4) Downward BLUE Mouse : (3,4)	[INVALID MOVE]	[INVALID MOVE]	Pass
15	can't capture enemy elephant while in river	BLUE Team (4,5) to the left RED Elephant : (4,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) Upward BLUE Elephant : (0,5)	[INVALID MOVE]	[INVALID MOVE]	Pass
16	can't capture mouse in river while on land	BLUE Team (3,4) Upward RED Mouse : (2,4)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (3,5) Upward BLUE Mouse : (2,5)	[INVALID MOVE]	[INVALID MOVE]	Pass
17	will capture higher ranked animal that is trapped	BLUE Team (3,2) to the Left RED Wolf : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (4,7) Upward BLUE Cat : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass

18	will not be able to capture higher ranked animal	BLUE Team (3,3) to the Left RED Wolf : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (4,5) Upward BLUE Cat : (3,5)	[INVALID MOVE]	[INVALID MOVE]	Pass
19	will reach the enemy's den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		REDTeam (4,0) Upward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : cat

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	can't move to own den	BLUE Team (2,0) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,8) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,5) to the Left	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
3	can go to it's own trap	BLUE Team (2,1) Downward	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,6) to the Left	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
4	can't go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass

5	can’t land on position occupied by teammate	BLUE Team (4,2) Upward BLUE Wolf : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Downward RED Wolf : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can’t move to river	BLUE Team (6,3) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
7	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Cat : (3,4)	cat will be captured	cat was captured	Pass
		RED Team (3,4) to the Left BLUE Cat : (3,3)	cat will be captured	cat was captured	Pass
8	will be captured by enemy with equal rank	BLUE Team (3,3) RED Cat : (3,4) to the Left	cat will be captured	cat was captured	Pass
		RED Team (3,4) BLUE Cat : (3,3) to the Right	cat will be captured	cat was captured	Pass
9	will capture trapped enemy	BLUE Team (3,2) to the Left RED Elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (2,7) Downward BLUE Elephant : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass
10	will be trapped on enemy’s trap and captured	BLUE Team (4,8) RED Mouse : (4,7) to the Right	cat will be captured	cat was captured	Pass
		RED Team (3,1)	cat will be captured	cat was captured	Pass

		BLUE Mouse : (3,2) to the Left			
11	can’t capture enemy with higher rank	BLUE Team (3,2) to the Right RED Tiger : (3,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Upward BLUE Tiger : (1,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
12	will capture enemy with lower rank	BLUE Team (4,2) Upward RED Mouse : (3,2)	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,6) to the Left BLUE Mouse : (3,5)	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
13	will reach the enemy’s den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		RED Team (2,0) Downward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : wolf

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	can’t move to own den	BLUE Team (2,0) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,8) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,5) to the Left	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
3	can go to it’s own trap	BLUE Team (2,1) Downward	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team	New Row = 3	New Row = 3	Pass

		(3,6) to the Left	New Column = 5	New Column = 5	
4	can’t go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
5	can’t land on position occupied by teammate	BLUE Team (4,2) Upward BLUE Wolf : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Downward RED Wolf : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can’t move to river	BLUE Team (6,3) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
7	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Wolf : (3,4)	wolf will be captured	wolf was captured	Pass
		RED Team (3,4) to the Left BLUE Wolf : (3,3)	wolf will be captured	wolf was captured	Pass
8	will be captured by enemy with equal rank	BLUE Team (3,3) RED Wolf : (3,4) to the Left	wolf will be captured	wolf was captured	Pass
		RED Team (3,4) BLUE Wolf : (3,3) to the Right	wolf will be captured	wolf was captured	Pass
9	will capture trapped enemy	BLUE Team (3,2) to the Left RED Elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (2,7) Downward BLUE Elephant : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass

10	will be trapped on enemy's trap and captured	BLUE Team (4,8) RED Mouse : (4,7) to the Right	cat will be captured	cat was captured	Pass
		RED Team (3,1) BLUE Mouse : (3,2) to the Left	cat will be captured	cat was captured	Pass
11	can't capture enemy with higher rank	BLUE Team (3,2) to the Right RED Tiger : (3,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Upward BLUE Tiger : (1,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
12	will capture enemy with lower rank	BLUE Team (4,2) Upward RED Mouse : (3,2)	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,6) to the Left BLUE Mouse : (3,5)	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
13	will reach the enemy's den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		RED Team (2,0) Downward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : dog

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	can't move to own den	BLUE Team (2,0) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,8) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can move to land	BLUE Team	New Row = 3	New Row = 3	Pass

		(3,1) to the Right	New Column = 2	New Column = 2	
		RED Team (3,5) to the Left	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
3	can go to it’s own trap	BLUE Team (2,1) Downward	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,6) to the Left	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
4	can’t go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
5	can’t land on position occupied by teammate	BLUE Team (4,2) Upward BLUE Wolf : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Downward RED Wolf : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can’t move to river	BLUE Team (6,3) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
7	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Dog : (3,4)	dog will be captured	dog was captured	Pass
		RED Team (3,4) to the Left BLUE Dog : (3,3)	dog will be captured	dog was captured	Pass
8	will be captured by enemy with equal rank	BLUE Team (3,3) RED Dog : (3,4) to the Left	dog will be captured	dog was captured	Pass
		RED Team (3,4) BLUE Dog : (3,3) to the Right	dog will be captured	dog was captured	Pass

9	will capture trapped enemy	BLUE Team (3,2) to the Left RED Elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (2,7) Downward BLUE Elephant : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass
10	will be trapped on enemy's trap and captured	BLUE Team (4,8) RED Mouse : (4,7) to the Right	cat will be captured	cat was captured	Pass
		RED Team (3,1) BLUE Mouse : (3,2) to the Left	cat will be captured	cat was captured	Pass
11	can't capture enemy with higher rank	BLUE Team (3,2) to the Right RED Tiger : (3,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Upward BLUE Tiger : (1,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
12	will capture enemy with lower rank	BLUE Team (4,2) Upward RED Mouse : (3,2)	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,6) to the Left BLUE Mouse : (3,5)	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
13	will reach the enemy's den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		RED Team (2,0) Downward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : leopard

	Test Description	**Input value/ current situation**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**

1	can’t move to own den	BLUE Team (2,0) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,8) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,5) to the Left	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
3	can go to it’s own trap	BLUE Team (2,1) Downward	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,6) to the Left	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
4	can’t go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
5	can’t land on position occupied by teammate	BLUE Team (4,2) Upward BLUE Wolf : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Downward RED Wolf : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can’t move to river	BLUE Team (6,3) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
7	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Leopard : (3,4)	leopard will be captured	leopard was captured	Pass
		RED Team (3,4) to the Left BLUE Leopard : (3,3)	leopard will be captured	leopard was captured	Pass
8	will be captured by	BLUE Team	leopard will be captured	leopard was captured	Pass

	enemy with equal rank	(3,3) RED Leopard : (3,4) to the Left			
		RED Team (3,4) BLUE Leopard (3,3) to the Right	leopard will be captured	leopard was captured	Pass
9	will capture trapped enemy	BLUE Team (3,2) to the Left RED Elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (2,7) Downward BLUE Elephant : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass
10	will be trapped on enemy's trap and captured	BLUE Team (4,8) RED Mouse : (4,7) to the Right	leopard will be captured	leopard was captured	Pass
		RED Team (3,1) BLUE Mouse : (3,2) to the Left	leopard will be captured	leopard was captured	Pass
11	can't capture enemy with higher rank	BLUE Team (3,2) to the Right RED Tiger : (3,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,6) Upward BLUE Tiger : (1,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
12	will capture enemy with lower rank	BLUE Team (4,2) Upward RED Dog : (3,2)	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (3,6) to the Left BLUE Wolf : (3,5)	New Row = 3 New Column = 5	New Row = 3 New Column = 5	Pass
13	will reach the enemy's den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		RED Team	New Row = 3	New Row = 3	Pass

		(2,0) Downward	New Column = 0 RED Team wins	New Column = 0 RED Team wins	

Animal : tiger

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	crosses river horizontally	BLUE Team (2,2) to the Right	New Row = 2 New Column = 6	New Row = 2 New Column = 6	Pass
		RED Team (4,6) to the Left	New Row = 4 New Column = 2	New Row = 4 New Column = 6	Pass
2	crosses river vertically	BLUE Team (3,3) Upward	New Row = 0 New Column = 3	New Row = 0 New Column = 3	Pass
		RED Team (3,3) Downward	New Row = 6 New Column = 3	New Row = 6 New Column = 3	Pass
3	can't move to own den	BLUE Team (2,0) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (4,8) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
4	can move to land	BLUE Team (0,0) to the Right	New Row = 0 New Column = 1	New Row = 0 New Column = 1	Pass
		RED Team (6,8) to the Left	New Row = 6 New Column = 7	New Row = 6 New Column = 7	Pass
5	can go to it's own trap	BLUE Team (2,1) Downward	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (5,8) Upward	New Row = 4 New Column = 8	New Row = 4 New Column = 8	Pass
6	can't go out of bounds	BLUE Team (0,2) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (6,7) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass
7	can't land on position occupied by teammate	BLUE Team (0,1) to the Right BLUE elephant : (0,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team	[INVALID MOVE]	[INVALID MOVE]	Pass

		(6,7) to the Left RED elephant: (6,6)			
8	will be captured by enemy with higher rank	BLUE Team (1,2) RED Lion : (1,6) to the Left	tiger will be captured	tiger was captured	Pass
		RED Team (5,6) BLUE Lion : (5,2) to the Right	tiger will be captured	tiger was captured	Pass
9	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Tiger (3,4)	New Row =3 New Column = 4	New Row =3 New Column = 4	Pass
		RED Team (3,4) to the Left BLUE Tiger (3,3)	New Row =3 New Column = 3	New Row =3 New Column = 3	Pass
10	will be captured by enemy with equal rank	BLUE Team (3,3) RED Tiger (3,4) to the Left	tiger will be captured	tiger was captured	Pass
		RED Team (3,4) BLUE Tiger (3,3) to the Right	tiger will be captured	tiger was captured	Pass
11	will be trapped on enemy’s trap and captured by enemy with lower rank	BLUE Team (3,7) RED mouse : (2,6) Downward	tiger will be captured	tiger was captured	Pass
		RED Team (3,1) BLUE mouse : (3,2) to the Left	tiger will be captured	tiger was captured	Pass
12	will capture animal with lower rank	BLUE Team (3,7) Downward RED cat : (4,7)	New Row = 4 New Column = 7	New Row = 4 New Column = 7	Pass
		RED Team (3,1) Downward BLUE wolf : (4,1)	New Row = 4 New Column = 1	New Row = 4 New Column = 1	Pass

13	can’t cross the river since there’s a mouse in the river	BLUE Team (3,5) Upward RED mouse : (1, 5)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,2) to the Right BLUE mouse : (2,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
14	will capture higher ranked animal that is trapped	BLUE Team (2,1) Downward RED elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (4,7) Upward BLUE elephant :(3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass
15	will not be able to capture higher ranked animal	BLUE Team (2,2) Downward RED elephant : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (4,6) Upward BLUE elephant : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
16	will reach the enemy’s den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		RED Team (3,1) to the Left	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : lion

	Test Description	Input value/ current situation	Expected output/ result	Actual output/result	Pass/Fail
1	crosses river horizontally	BLUE Team (5,2) to the Right	New Row = 5 New Column = 6	New Row = 5 New Column = 6	Pass
		RED Team (2,5) to the Left	New Row = 2 New Column = 2	New Row = 2 New Column = 2	Pass
2	crosses river vertically	BLUE Team (6,3) Upward	New Row = 3 New Column = 3	New Row = 3 New Column = 3	Pass

		RED Team (0,4) Downward	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
3	can’t move to own den	BLUE Team (3,1) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (3,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
4	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (5,6) to the Left	New Row = 5 New Column = 5	New Row = 5 New Column = 5	Pass
5	can go to it’s own trap	BLUE Team (3,2) to the Left	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,5) to the Right	New Row = 3 New Column = 6	New Row = 3 New Column = 6	Pass
6	can’t go out of bounds	BLUE Team (0,6) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
7	can’t land on position occupied by teammate	BLUE Team (6,0) to the Right BLUE mouse : (6,1)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (0,7) to the Left RED mouse: (0,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
8	will be captured by enemy with higher rank	BLUE Team (1,2) RED Lion : (1,6) to the Left	tiger will be captured	tiger was captured	Pass
		RED Team (5,6) BLUE Lion : (5,2) to the Right	tiger will be captured	tiger was captured	Pass
9	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Tiger (3,4)	New Row =3 New Column = 4	New Row =3 New Column = 4	Pass

		RED Team (3,4) to the Left BLUE Tiger (3,3)	New Row =3 New Column = 3	New Row =3 New Column = 3	Pass
10	will be captured by enemy with equal rank	BLUE Team (3,3) RED Tiger (3,4) to the Left	tiger will be captured	tiger was captured	Pass
		RED Team (3,4) BLUE Tiger (3,3) to the Right	tiger will be captured	tiger was captured	Pass
11	will be trapped on enemy's trap and captured by enemy with lower rank	BLUE Team (3,7) RED mouse : (2,6) Downward	tiger will be captured	tiger was captured	Pass
		RED Team (3,1) BLUE mouse : (3,2) to the Left	tiger will be captured	tiger was captured	Pass
12	will capture animal with lower rank	BLUE Team (3,7) Downward RED cat : (4,7)	New Row = 4 New Column = 7	New Row = 4 New Column = 7	Pass
		RED Team (3,1) Downward BLUE wolf : (4,1)	New Row = 4 New Column = 1	New Row = 4 New Column = 1	Pass
13	can't cross the river since there's a mouse in the river	BLUE Team (3,5) Upward RED mouse : (1, 5)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,2) to the Right BLUE mouse : (2,3)	[INVALID MOVE]	[INVALID MOVE]	Pass
14	will capture higher ranked animal that is trapped	BLUE Team (2,1) Downward RED elephant : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (4,7) Upward BLUE elephant :(3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass

15	will not be able to capture higher ranked animal	BLUE Team (2,2) Downward RED elephant : (3,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (4,6) Upward BLUE elephant : (3,6)	[INVALID MOVE]	[INVALID MOVE]	Pass
16	will reach the enemy’s den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		REDTeam (4,0) Upward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

Animal : elephant

	Test Description	**Input value/ current situation**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	can’t cross river	BLUE Team (6,2) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (1,5) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
2	can’t move to own den	BLUE Team (3,1) to the Left	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (3,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
3	can move to land	BLUE Team (3,1) to the Right	New Row = 3 New Column = 2	New Row = 3 New Column = 2	Pass
		RED Team (6,6) to the Left	New Row = 6 New Column = 5	New Row = 6 New Column = 5	Pass
4	can go to it’s own trap	BLUE Team (3,2) to the Left	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (3,5) to the Right	New Row = 3 New Column = 6	New Row = 3 New Column = 6	Pass
5	can’t go out of bounds	BLUE Team (6,1) Downward	[INVALID MOVE]	[INVALID MOVE]	Pass

		RED Team (2,7) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
6	can’t land on position occupied by teammate	BLUE Team (6,1) to the Left BLUE Lion : (6,0)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (2,7) Upward RED leopard: (1,7)	[INVALID MOVE]	[INVALID MOVE]	Pass
7	will be captured by mouse	BLUE Team (1,2) RED Mouse : (0,2) Downward	the elephant will be captured	the elephant will be captured	Pass
		RED Team (6,5) BLUE Mouse : (6,4) to the Right	the elephant will be captured	the elephant will be captured	Pass
8	will capture enemy with equal rank	BLUE Team (3,3) to the Right RED Elephant : (3,4)	New Row = 3 New Column = 4	New Row = 3 New Column = 4	Pass
		RED Team (3,4) to the Left BLUE Elephant : (3,3)	New Row = 3 New Column = 3	New Row = 3 New Column = 3	Pass
9	will be captured by enemy with equal rank	BLUE Team (3,3) RED Elephant : (3,4) to the Left	elephant will be captured	elephant will be captured	Pass
		RED Team (3,4) BLUE Elephant (3,3) to the Right	elephant will be captured	elephant will be captured	Pass
10	will be trapped on enemy’s trap and captured by enemy	BLUE Team (3,7) RED Wolf : (3,6) to the Right	elephant will be captured	elephant will be captured	Pass
		RED Team (3,1) BLUE Leopard : (4,1) Upward	elephant will be captured	elephant will be captured	Pass

11	will capture mouse that is trapped	BLUE Team (2,1) Downward RED Mouse : (3,1)	New Row = 3 New Column = 1	New Row = 3 New Column = 1	Pass
		RED Team (2,7) Downward BLUE Mouse : (3,7)	New Row = 3 New Column = 7	New Row = 3 New Column = 7	Pass
12	can’t move to river	BLUE Team (1,2) to the Right	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (6,5) Upward	[INVALID MOVE]	[INVALID MOVE]	Pass
13	will not be able to capture an enemy’s mouse	BLUE Team (1,2) Upward RED Mouse : (0,2)	[INVALID MOVE]	[INVALID MOVE]	Pass
		RED Team (6,5) to the Left BLUE Mouse : (6,4)	[INVALID MOVE]	[INVALID MOVE]	Pass
14	will reach the enemy’s den	BLUE Team (3,7) to the Right	New Row = 3 New Column = 8 BLUE Team wins	New Row = 3 New Column = 8 BLUE Team wins	Pass
		REDTeam (4,0) Upward	New Row = 3 New Column = 0 RED Team wins	New Row = 3 New Column = 0 RED Team wins	Pass

FOLDER : MODEL | CLASS : GAME

Folder Model
Class : Game
Method: public static void setPlayer (String strName)
Method Description: Set a Player and add is into the arrPlayers

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	arrPlayers is still empty	strName = sampleA	create a new player named sampleA and store it to arrPlayers	create a new player named sampleA and store it to arrPlayers	Pass
2	arrPlayers already has 1 size()	strName = sampleB	create a new player named sampleB and store it to arrPlayers	create a new player named sampleB and store it to arrPlayers	Pass
3	arrPlayers already has 2 or more size()	strName = sampleC	no player will be created and not be stored in arrPlayers	no player will be created and not be stored in arrPlayers	Pass

Folder Model
Class : Game
Method: public int draw ()
Method Description: Returns a randomized number from 0-7

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player 1 draws a number for the first time		Returns a randomized number from 0-7	Returns a randomized number from 0-7	Pass
2	player 2 draws a number for the first time		Returns a randomized number from 0-7	Returns a randomized number from 0-7	Pass
3	player 1 draws a number for the second time		Returns a randomized number from 0-7	Returns a randomized number from 0-7	Pass
4	player 2 draws a number for the second time		Returns a randomized number from 0-7	Returns a randomized number from 0-7	Pass

Folder Model
Class : Game

Method:public void pickColor (boolean bPlayerInTurn, boolean bBlue)
Method Description: Set the first player to attack to chosen team color he/she wanted

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player 1 picks color blue	bPlayerInTurn = true bBlue = true	player 1 set color as blue	player 1 set color as blue	Pass
2	player 2 picks color blue	bPlayerInTurn = false bBlue = true	player 2 set color as blue	player 2 set color as blue	Pass
3	player 1 picks color red	bPlayerInTurn = true bBlue = false	player 1 set color as red	player 1 set color as red	Pass
4	player 2 picks color red	bPlayerInTurn = false bBlue = false	player 2 set color as red	player 2 set color as red	Pass

Folder Model
Class : Game
Method: public void setPlayerInTurn (boolean bBlue)
Method Description: Set the current player in turn

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Set Player in turn to blue	bBlue = true	Set Player in turn to blue	Set Player in turn to blue	Pass
2	Set Player in turn to red	bBlue = false	Set Player in turn to red	Set Player in turn to red	Pass

Folder Model
Class : Game
Method: public static int getMAX_PLAYERS()
Method Description: Get count of Players

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Get count of Players		return int count of Players	return int count of Players	Pass

Folder Model
Class : Game
Method: public static int getMAX_ANIMALS()
Method Description: Get the count of ALL animals both blue and not

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Get the count of ALL animals both blue and not		return count of ALL animals both blue and not	return count of ALL animals both blue and not	Pass

Folder Model
Class : Game
Method: public static boolean getPlayerTurn ()
Method Description: Get the count of ALL animals both blue and not

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Get the current color of the Player In Turn when player in turn is blue		return true	return true	Pass
2	Get the current color of the Player In Turn when player in turn is red		return false	return false	Pass

Folder Model
Class : Game
Method: public BoardGame getBoardGame()
Method Description: Get the whole BoardGame

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Get the whole BoardGame		return BoardGame bBoardGame = board of the game	return BoardGame bBoardGame = board of the game	Pass

Folder Model
Class : Game
Method: public static Player getPlayer (String strName)
Method Description: Get the player based on their name

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	get the player A based on name	strName = playerAName	return player with name playerAName	return player with name playerAName	Pass
2	get the player B based on name	strName = playerBName	return player with name playerBName	return player with name playerBName	Pass

Folder: Model
Class : Game
Method: public static Player getPlayer (boolean bBlue)
Method Description: Get the Player based on their team

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	get the player A color blue	bBlue = true	return player with color blue	return player with color blue	Pass
2	get the player B color red	bBlue = false	return player with color blue	return player with color blue	Pass

Folder: Model
Class : Game
Method: public static Player getPlayer (int nIndex)
Method Description: Get the player based on the players index on the arraylist

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	get the player A in index 0 of the arrPlayers	nIndex = 0	return the player A in index 0 of the arrPlayers	return the player A in index 0 of the arrPlayers	Pass
2	get the player A in index 1 of the arrPlayers	nIndex = 1	return the player A in index 1 of the arrPlayers	return the player A in index 1 of the arrPlayers	Pass

Folder: Model
Class : Game
Method:public boolean isNoWinner ()
Method Description: check if all player is still not a winner

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	There is already a winner	nIndex = 0	return true	return true	Pass
2	there is no winner yet	nIndex = 1	return false	return false	Pass

Folder: Model
Class : Game
Method: public String displayWinner ()
Method Description: Display the name of the winning player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	arrPlayer[0] is the winner		return the name of arrPlayer[0]	return the name of arrPlayer[0]	Pass
2	arrPlayer[1] is the winner		return the name of arrPlayer[1]	return the name of arrPlayer[1]	Pass

Folder: Model
Class : Game
Method: public boolean moveAnimal (int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Move the animal in the INITIAL ROW and COL to POST ROW and COL

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the move is not possible	cat will go to the river	return false	return false	Pass
2	the parameters are invalid	int nInitX = -1, int nInitY= -1, int nPostX= -1, int nPostY= -1	return false	return false	Pass
3	able to move	mouse will eat the elephant one cell away	return true	return true	Pass

Folder: Model
Class : Game
Method: private boolean updateStatus (int nPostX, int nPostY)
Method Description: Update the status of the move Animals if captured or not, and Checks if the current player in turn already won

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	move out the animal from the trap but not Den	red animal goes out of trap	Set animal as NOT captured return false	Set animal as NOT captured return false	Pass
2	move in the animal to the trap which is not Den	red animal goes to blue den	Set animal as captured return false	Set animal as captured return false	Pass
3	move from not trap to not trap	blue animal from not trap to not trap	Set animal as NOT captured return false	Set animal as NOT captured return false	Pass
4	move from trap to Den	blue animal from trap to red den	Set animal as NOT captured return true	Set animal as NOT captured return true	Pass

Folder: Model
Class : Game
Method: private boolean isAllEnemyKilled ()
Method Description: Check if all enemy of the current player in turn is already killed

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	red player already eat all blue animals		return true	return true	Pass
2	red player not yet eat all blue animals		return false	return false	Pass
3	blue player already eat all red animals		return true	return true	Pass
4	blue player NOT yet eat all red animals		return false	return false	Pass

Folder: Model
Class : Game
Method: public void changePlayerInTurn ()
Method Description: Changes the Player In Turn

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	current player in turn is red		bPlayerInTurn = true	bPlayerInTurn = true	Pass
2	current player in turn is blue		bPlayerInTurn = false	bPlayerInTurn = false	Pass
3	current player in turn is red		bPlayerInTurn = true	bPlayerInTurn = true	Pass

Folder: Model
Class : Game
Method: public void pushArrCells()
Method Description: Save the current state of the arrCells in the bBoardGame for the undo History

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	push for the first time		store the first state of arrCells	store the first state of arrCells	Pass
2	push for the second time		store the second state of arrCells	store the second state of arrCells	Pass
3	push for the 10 time		store the 10 state of arrCells	store the 10state of arrCells	Pass

Folder: Model
Class : Game
Method: public boolean isEmptyHistory()
Method Description: Checks if there is no previous state stored to be undone

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	undoStackHistory is empty		return true	return true	Pass
2	undoStackHistory has 1 stack		return false	return false	Pass
3	undoStackHistory has 10 stack		return false	return false	Pass

Folder: Model
Class : Game
Method: public boolean isNoMove(int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Check if the move is placed to same row and column

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	same initial position and post	int nInitX = 0, int nInitY= 0, int nPostX= 0, int nPostY= 0	return true	return true	Pass
2	different col same row	int nInitX= 0, int nInitY= 0, int nPostX= 0, int nPostY= 1	return false	return false	Pass
3	different row sam col	int nInitX= 1, int nInitY= 0, int nPostX= 0, int nPostY= 0	return false	return false	Pass

Folder: Model
Class : Game
Method: public void undo ()
Method Description: Revert back to the previous state arrCells (especially the animals)

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	undo when the left stack is the original state		Do not undo the game	Do not undo the game	Pass
2	undoStackHistory has 2 stacks		Do undo the game	Do undo the game	Pass
3	undoStackHistory has 10 stacks		Do undo the game	Do undo the game	Pass

Folder: Model
Class : Game
Method: public void redo()
Method Description: Revert back to the undone state arrCells (especially the animals)

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	redowhen the left stack is the latest state		Do not redo the game	Do not redo the game	Pass
2	redoStackHistory has 2 stacks		Do redo game	Do redo the game	Pass
3	redoStackHistory has 10 stacks		Do redo the game	Do redo the game	Pass

Folder: Model
Class : Game
Method: public void resetGame()
Method Description: Reset the game to its original state

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	reset game when a player already won by going to enemy’s den		resets the game	resets the game	Pass
2	reset game when a player quits the game		resets the game	resets the game	Pass
3	reset game when a player already won through killing all animals		resets the game	resets the game	Pass

FOLDER : MODEL | CLASS : BOARDGAME

Folder: Model
Class : BoardGame
Method: public BoardCell[][] cloneBoardCells()
Method Description: Clones the current version of Animals in arrCells

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Blue Team’s Tiger moved to the right from initial position		arrCells cloned	arrCells cloned	Pass
2	Red Team’s Lion moved to the left from initial position		arrCells cloned	arrCells cloned	Pass
3	Blue Team’s Dog moved upward from initial position		arrCells cloned	arrCells cloned	Pass

Folder: Model
Class : BoardGame
Method: public void pushArrCellsForUndo()
Method Description: Push the current version of arrCells to undoStackHistory

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Blue Team’s Tiger moved to the right from initial position		arrCells pushed to undoStackHistory	arrCells pushed to undoStackHistory	Pass
2	Red Team’s Lion moved to the left from initial position		arrCells pushed to undoStackHistory	arrCells pushed to undoStackHistory	Pass
3	Blue Team’s Dog moved upward from initial position		arrCells pushed to undoStackHistory	arrCells pushed to undoStackHistory	Pass

Folder: Model
Class : BoardGame
Method: public boolean undo ()
Method Description: Undoes the Animals in the GameBoard

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Blue Team's Tiger moved to the right from initial position		will return true	returned true	Pass
2	No Animal has moved yet		will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: public boolean redo ()
Method Description: Redoes the Animals in the GameBoard

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Blue Team's Tiger moved to the right from initial position		will return true	returned true	Pass
2	No Animal has moved yet		will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: public boolean isUndoStackHistoryEmpty()
Method Description: Checks if the undoStackHistory is Empty

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	Blue Team's Tiger moved to the right from initial position		will return false	returned false	Pass
2	No Animal has moved yet		will return true	returned true	Pass

Folder: Model

Class : BoardGame
Method: public void resetBoardGame()
Method Description: Reset the arrCells animals to its original state

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	reset game when a player already won by going to enemy’s den		resets the arrCells	resets the arrCells	Pass
2	reset game when a player quits the game		resets the arrCells	resets the arrCells	Pass
3	reset game when a player already won through killing all animals		resets the arrCells	resets the arrCells	Pass

Folder: Model
Class : BoardGame
Method: public void resetBoardGame()
Method Description: Reset the arrCells animals to its original state

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	reset game when a player already won by going to enemy’s den		resets the arrCells	resets the arrCells	Pass
2	reset game when a player quits the game		resets the arrCells	resets the arrCells	Pass
3	reset game when a player already won through killing all animals		resets the arrCells	resets the arrCells	Pass

Folder: Model
Class : BoardGame
Method: private void setTraps ()
Method Description: Set the arrCells trap Locations and color

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	to set Blue Team's traps		Blue Team's traps will be set	Blue Team's traps have been set	Pass
2	to set Red Team's traps		Red Team's traps will be set	Red Team's traps have been set	Pass

Folder: Model
Class : BoardGame
Method: private void setAnimals ()
Method Description: Create animals to the first locations

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	to set the initial locations of Blue Team's animals		locations of Blue Team's animals will be set	locations of Blue Team's animals have been set	Pass
2	to set the initial locations of Red Team's animals		locations of Red Team's animals will be set	locations of Red Team's animals have been set	Pass

Folder: Model
Class : BoardGame
Method: public static int getMAX_PLAYERS ()
Method Description: Get the MAX_PLAYERS

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	gets the MAX_PLAYERS		will return 2	returned 2	Pass

Folder: Model
Class : BoardGame
Method: public static int getMAX_ANIMALS ()
Method Description: Gets the MAX_ANIMALS

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	gets the MAX_ANIMALS		will return 16	returned 16	Pass

Folder: Model
Class : BoardGame
Method: public static int getMAX_ROW ()
Method Description: Gets the MAX_ROW

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	gets the MAX_ROW		will return 7	returned 7	Pass

Folder: Model
Class : BoardGame
Method: public static int getMAX_COL ()
Method Description: Gets the MAX_COL

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	gets the MAX_COL		will return 9	returned 9	Pass

Folder: Model
Class : BoardGame
Method: public BoardCell getBoardCell (int nX, int nY)
Method Description: Gets the specific BoardCell in BoardCell[][]

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	nX and nY are out of bounds	nX = -1 nY = -1	will return null	returned null	Pass
2	nX and nY are not out of bounds	nX = 4 nY = 4	will return arrCells[4][4]	returned arrCells[4][4]	Pass

Folder: Model
Class : BoardGame
Method: public boolean moveAnimal (int nInitX, int nInitY, int nPostX, int nPostY, boolean bPlayerInTurn)
Method Description: Move the animal in the INITIAL ROW and COL to POST ROW and COL

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	a move is possible	nInitX = 4 nInitY = 8 nPostX = 4 nPostY = 7 bPlayerInTurn = false	will return true	returned true	Pass
2	a move is not possible	nInitX = 0 nInitY = 0 nPostX = -1 nPostY = -1 bPlayerInTurn = true	will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: public boolean isOutOfBounds(int nX, int nY)
Method Description: Check if the ROW and COL is Out of Bounds of the declared arrCells

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a row and column is out of bounds	nX = 8 nY = 8	will return true	returned true	Pass
2	a row and column is not out of bounds	nX = 0 nY = 0	will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: private boolean canEat(int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Checks if the Animal in the INITIAL ROW AND COL and eat the Animal in the POST ROW AND COL

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a mouse in the river tries to eat an enemy elephant on land	nInitX = 2 nInitY = 4 nPostX = 3 nPostY = 4 bPlayerInTurn = false	will return false	returned false	Pass

2	a mouse in the river tries to eat the enemy mouse in the river	nInitX = 2 nInitY = 4 nPostX = 2 nPostY = 5 bPlayerInTurn = true	will return true	returned true	Pass

Folder: Model
Class : BoardGame
Method: private boolean isProperDirection(int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Checks if there is no SLANTED move

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a move is diagonal	nInitX = 2 nInitY = 4 nPostX = 1 nPostY = 5	will return false	returned false	Pass
2	a move is not diagonal	nInitX = 2 nInitY = 4 nPostX = 2 nPostY = 5	will return true	returned true	Pass

Folder: Model
Class : BoardGame
Method: private boolean canAnimalPerform (int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Checks if the non abstract class of the animal can perform the inputted move from arrCells

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a leopard tries to jump across the river	nInitX = 2 nInitY = 7 nPostX = 2 nPostY = 2	will return false	returned false	Pass
2	a mouse in the river tries to eat the enemy mouse in the river	nInitX = 2 nInitY = 4	will return true	returned true	Pass

		nPostX = 2 nPostY = 5			
3	a mouse in the river tries to eat an enemy elephant on land	nInitX = 2 nInitY = 4 nPostX = 2 nPostY = 5	will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: private boolean isSingleCellMove (int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Checks if the move only moves a distance of 1 arrCells

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the distance between initial location and post location is 1	nInitX = 4 nInitY = 4 nPostX = 3 nPostY = 4	will return true	returned true	Pass
2	the distance between initial location and post location is 3	nInitX = 2 nInitY = 2 nPostX = 2 nPostY = 7	will return false	returned false	Pass
3	the distance between initial location and post location is 0	nInitX = 4 nInitY = 4 nPostX = 4 nPostY = 4	will return false	returned false	Pass

Folder: Model
Class : BoardGame
Method: private boolean isPossibleJump (int nInitX, int nInitY, int nPostX, int nPostY)
Method Description: Checks if the Jump is done properly in the river and without mouse in between

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a jump is possible	nInitX = 0 nInitY = 3	will return true	returned true	Pass

		nPostX = 3 nPostY = 3			
2	a jump is not possible	nInitX = 2 nInitY = 3 nPostX = 2 nPostY = 7	will return false	returned false	Pass

Folder: Model

Class : BoardGame

Method: public boolean updateStatus (int nPostX, int nPostY, boolean bPlayerInTurn)

Method Description: Updates the isCaptured animal in the POST ROW AND COL AND checks if the currentPlayerInTurn is able to go to enemy's Den

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a BLUE mouse moved to the RED Team's den	nPostX = 3 nPostY = 8 bPlayerInTurn = true	will return true	returned true	Pass
2	a BLUE mouse did not move to the RED Team's den	nPostX = 2 nPostY = 4 bPlayerInTurn = true	will return false	returned false	Pass

Folder: Model

Class : BoardGame

Method: public boolean isCheckMate (boolean bPlayerInTurn)

Method Description: Check if the current playerInTurn has no possible moves left

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player has possible moves left	bPlayerInTurn = true	will return true	returned true	Pass
2	a player doesn't have any possible moves left	bPlayerInTurn = true	will return false	returned false	Pass

FOLDER : MODEL | CLASS : PLAYER

Folder: Model
Class : Player
Method: public void setName(String strName)
Method Description: Set the name of the Player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	name of player 1 is set	strName = “Kriz”	strName is set	strName is set	Pass
2	name of player 2 is set	strName = “Faith”	strName is set	strName is set	Pass

Folder: Model
Class : Player
Method: public void setKilledAnimalCounter(int nKilledEnemyAnimalCtr)
Method Description:Sets the Killed Enemy Animal count of the Player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player 1 has no kills yet	nKilledEnemyAnimalCtr = 0	nKilledEnemyAnimalCtr is set	nKilledEnemyAnimalCtr is set	Pass
2	player 2 killed 2 animals	nKilledEnemyAnimalCtr = 2	nKilledEnemyAnimalCtr is set	nKilledEnemyAnimalCtr is set	Pass

Folder: Model
Class : Player
Method: public String getName()
Method Description:Gets the name of the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	get name of player 1		will return “Kriz”	returned “Kriz”	Pass
2	get name of player 1		will return “Faith”	returned “Faith”	Pass

Folder: Model
Class : Player
Method:public boolean isWinner ()
Method Description: Gets bWInner state of the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player 1 has is not the winner		will return false	returned false	Pass
2	player 2 is the winner		will return true	returned true	Pass

Folder: Model
Class : Player
Method:public void incrementKilledEnemyAnimalCounter ()
Method Description: Add 1 to the Killed Enemy Animal count of the Player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player 1 killed an enemy animal for the first time		nKilledEnemyAnimalCtr = 1	nKilledEnemyAnimalCtr = 1	Pass

Folder: Model
Class : Player
Method:public void resetPlayer()
Method Description: Delete all the information about the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	reset player when a player already won by going to enemy's den		resets the player	resets the player	Pass
2	reset player when a player quits the game		resets the player	resets the player	Pass
3	reset player when a player already won through killing all animals		resets the player	resets the player	Pass

Folder: Model
Class : Player
Method: public void setColor (boolean bBlue)
Method Description: Set the team color of the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player's team color is set to BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	a player's team color is set to RED	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Player
Method: public boolean isBlue()
Method Description: Checks if the color of the class is Blue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player's team color is BLUE		bBlue will be set to true	bBlue was set to true	Pass
2	a player's team color is RED		bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Player
Method:public boolean isOwnColor(boolean bBlue)
Method Description: Checks if the color of the class is the same with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player's team color is BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	a player's team color is BLUE	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Player
Method:public boolean isEnemyColor(boolean bBlue)
Method Description: Checks if the color of the class is the different with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player's team color is BLUE	bBlue = false	bBlue will be set to true	bBlue was set to true	Pass
2	a player's team color is BLUE	bBlue = true	bBlue will be set to false	bBlue was set to false	Pass

FOLDER : MODEL | CLASS : ANIMAL

Folder: Model
Class : Animal
Method: public void setRank(int nRank)
Method Description: This function sets the rank of an Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the rank of an instance of Animal child is 4	nRank = 4	nRank will be set	nRank was set	Pass

Folder: Model
Class : Animal
Method: public void setCanJump(boolean bCanJump)
Method Description: This function sets whether an Animal object can jump

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the child of Animal can jump	bCanJump = true	bCanJump will be set	bCanJump was set	Pass
2	the child of Animal can't jump	bCanJump = false	bCanJump will be set	bCanJump was set	Pass

Folder: Model
Class : Animal
Method: public void setCanWater(boolean bCanWater)
Method Description: This function sets whether an Animal object can walk in the river

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	the child of Animal can walk in the river	bCanWater = true	bCanWater will be set	bCanWater was set	Pass
2	the child of Animal can’t walk in the river	bCaWater = false	bCanWater will be set	bCanWater was set	Pass

Folder: Model
Class : Animal
Method: public void setIsAlive(boolean bAlive)
Method Description: This function sets the bAlive attribute of an Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the instance of a child of Animal can walk is alive	bAlive = true	bAlive will be set	bAlive was set	Pass
2	the instance of a child of Animal is not alive	bAlive = false	bAlive will be set	bAlive was set	Pass

Folder: Model
Class : Animal
Method: public int getRank()
Method Description: This function returns the rank of an Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the rank of an instance of Animal child is 4		will return 4	returned 4	Pass

Folder: Model
Class : Animal
Method: public boolean getCanJump()
Method Description: This function returns the bCanJump attribute of an Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an instance of Animal child can’t jump		will return false	returned false	Pass

2	an instance of Animal child can jump		will return true	returned true	Pass

Folder: Model
Class : Animal
Method: public boolean getCanWater()
Method Description: This function returns the bCanWater attribute of an Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an instance of Animal child can’t move into the rivers		will return false	returned false	Pass
2	an instance of Animal child can move into the rivers		will return true	returned true	Pass

Folder: Model
Class : Animal
Method: public boolean isAlive()
Method Description: This function checks if an animal is still alive.

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an instance of Animal child is not alive		will return false	returned false	Pass
2	an instance of Animal child is alive		will return true	returned true	Pass

Folder: Model
Class : Animal
Method: public boolean isCaptured()
Method Description: This function checks if an Animal object is captured

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	an instance of Animal child is not captured		will return false	returned false	Pass
2	an instance of Animal child is captured		will return true	returned true	Pass

Folder: Model
Class : Animal
Method: public boolean isLowerOrEqualRank(int nRank)
Method Description: This function checks if the rank of an Animal object is lower or equal to another Animal object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an instance of Animal child has a higher rank than another instance of Animal child	this.nRank = 4 nRank = 1	will return true	returned true	Pass
2	an instance of Animal child has a lower rank than another instance of Animal child	this.nRank = 3 nRank = 4	will return false	returned false	Pass

Folder: Model
Class : Animal
Method: public void setColor (boolean bBlue)
Method Description: Set the team color of the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an Animal’s team color is set to BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	an Animal’s team color is set to RED	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Animal
Method: public boolean isBlue()

Method Description: Checks if the color of the class is Blue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an Animal’s team color is BLUE		bBlue will be set to true	bBlue was set to true	Pass
2	an Animal’s team color is RED		bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Animal
Method:public boolean isOwnColor(boolean bBlue)
Method Description: Checks if the color of the class is the same with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an Animal's team color is BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	an Animal’s team color is BLUE	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : Animal
Method:public boolean isEnemyColor(boolean bBlue)
Method Description: Checks if the color of the class is the different with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	an Animal’s team color is BLUE	bBlue = false	bBlue will be set to true	bBlue was set to true	Pass
2	an Animal’s team color is BLUE	bBlue = true	bBlue will be set to false	bBlue was set to false	Pass

FOLDER : MODEL | CLASS : BOARDCELL

Folder: Model
Class : BoardCell
Method:public void setAnimal (Animal aAnimal)
Method Description: Set the aAnimal as the new animal located to this BoardCell

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	setAnimal() is called		a Boardcell object will be assigned an animal	a Boardcell object was assigned an animal	Pass

Folder: Model
Class : BoardCell
Method:public void setRiver ()
Method Description: Set the River. If called will set the bRiver to true

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	setRiver() is called		a Boardcell object will be assigned as a River	a Boardcell object was assigned as a River	Pass

Folder: Model
Class : BoardCell
Method:public void setDen ()
Method Description: Set the Den. If called will set the bDen to true

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	setDen() is called		a Boardcell object will be assigned as a Den	a Boardcell object was assigned as a Den	Pass

Folder: Model
Class : BoardCell
Method:public void setTrap ()
Method Description: Set the Trap. If called will set the bTrap to true

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	setTrap() is called		a Boardcell object will be assigned as a Trap	a Boardcell object was assigned as a Trap	Pass

Folder: Model
Class : BoardCell
Method:public Animal getAnimal()
Method Description: Gets an Animal

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a BoarCell object has an Animal		will return the Animal	returned the Animal	Pass
2	a BoarCell object doesn't have an Animal		will return null	returned null	Pass

Folder: Model
Class : BoardCell
Method:public boolean isRiver ()
Method Description: Check if the BoardCell is a River

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a BoarCell is a River		will return true	returned true	Pass
2	a BoarCell is not a River		will return false	returned false	Pass

Folder: Model
Class : BoardCell
Method:public boolean isTrap ()
Method Description: Check if the BoardCell is a Trap

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a BoarCell is a Trap		will return true	returned true	Pass
2	a BoarCell is not a Trap		will return false	returned false	Pass

Folder: Model

Class : BoardCell
Method:public boolean isDen ()
Method Description: Check if the BoardCell is a Den

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a BoarCell is a Den		will return true	returned true	Pass
2	a BoarCell is not a Den		will return false	returned false	Pass

Folder: Model
Class : BoardCell
Method: public void setColor (boolean bBlue)
Method Description: Set the team color of the player

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a Den's color is set to BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	a Den's color is set to RED	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : BoardCell
Method: public boolean isBlue()
Method Description: Checks if the color of the class is Blue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a Den's team color is BLUE		bBlue will be set to true	bBlue was set to true	Pass
2	a Den's team color is RED		bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : BoardCell
Method:public boolean isOwnColor(boolean bBlue)
Method Description: Checks if the color of the class is the same with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a Den's color is BLUE	bBlue = true	bBlue will be set to true	bBlue was set to true	Pass
2	a Den's color is BLUE	bBlue = false	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : BoardCell
Method:public boolean isEnemyColor(boolean bBlue)
Method Description: Checks if the color of the class is the different with bBlue

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a Den's color is BLUE	bBlue = false	bBlue will be set to true	bBlue was set to true	Pass
2	a Den's color is BLUE	bBlue = true	bBlue will be set to false	bBlue was set to false	Pass

Folder: Model
Class : BoardCell
Method:public void removeAnimal ()
Method Description: Set the Animal in the BoardCell as null

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	removeAnimal() is called		the Animal in a BoardCell will be removed	the Animal in a BoardCell was removed	Pass

FOLDER : MODEL | CLASS : MODELMAINMENU

Folder: Model
Class : ModelMainMenu
Method:public Game getGame()
Method Description:This function returns the game attribute

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	getGame() is called		an instance of Game will be returned	an instance of Game was returned	Pass

Folder: Model

Class : ModelMainMenu
Method:public String getStrInstructions()
Method Description:This function returns the strInstructions of a ModelMainMenu object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	getStrInstructions() is called		the instructions of the game will be returned	the instructions of the game was returned	Pass

Folder: Model
Class : ModelMainMenu
Method:public ModelSettings getSettings()
Method Description:This function returns the ModelSettings of a ModelMainMenu object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	getSettings() is called		the sound settings of the game will be returned	the sound settings of the game was returned	Pass

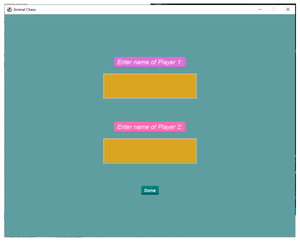
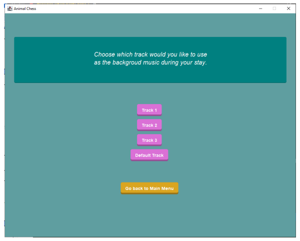
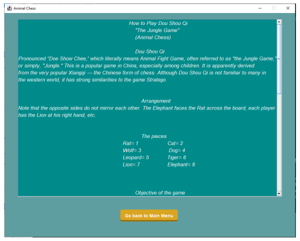
FOLDER : MODEL | CLASS : MODELSETTINGS

Folder: Model
Class : ModelSettings
Method:public File getFile(int nIndex)
Method Description:This function returns a File inside arrTracks of a ModelSettings object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	to obtain File of soundtrack 1	nIndex = 1	File of soundtrack 1 will be returned	File of soundtrack 1 was returned	Pass
2	to obtain File of default soundtrack	nIndex = 0	File of default soundtrack will be returned	File of default soundtrack was returned	Pass
3	to obtain File of soundtrack 2	nIndex = 2	File of soundtrack 2 will be returned	File of soundtrack 2 was returned	Pass

FOLDER : CONTROLLER | CLASS : GAMECONTROLLER

Folder: Controller
Class : GameController
Method:public void actionPerformed(ActionEvent e)
Method Description: This function listens to the GUI buttons with an ActionListener and specifies the activities to be done if a particular button is pressed

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	"Start Game" button was pressed		the setup game will be visible	 Animal Chess Enter name of Player 1 Enter name of Player 2 Done	Pass
2	"Sound Settings" button was pressed		the sound settings will be visible	 Choose which track would you like to use as the backgroud music during your stay. Track 1 Track 2 Track 3 Default Track Go back to Main Menu	Pass
3	"How to Play" button was pressed		the how to play will be visible	 How to Play Dou Shou Qi "The Jungle Game" (Animal Chess) Dou Shou Qi Pronounced "Doe Show Chee,' which literally means Animal Fight Game, often referred to as "the Jungle Game," or simply, "Jungle." This is a popular game in China, especially among children. It is apparently derived from the very popular Xiangqi — the Chinese form of chess. Although Dou Shou Qi is not familiar to many in the western world, it has strong similarities to the game Stratego. Arrangement Note that the opposite sides do not mirror each other. The Elephant faces the Rat across the board; each player has the Lion at his right hand, etc. The pieces Rat= 1 Cat= 2 Wolf= 3 Dog= 4 Leopard= 5 Tiger= 6 Lion= 7 Elephant= 8 Objective of the game 	Pass

Folder: Controller
Class : GameController
Method:public void updateAudioInputStream(File f)

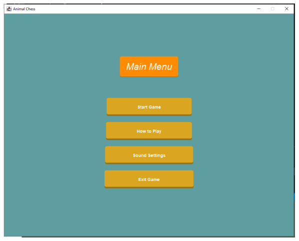
Method Description: This function gets the audio input stream given a File object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the updateAusioInputStream() is called		the Clip that is the audio will be set	the Clip that is the audio was set	Pass

Folder: Controller
Class : GameController
Method:public void controlSoundVolume(float f)
Method Description: This function controls the sound volume given a float value

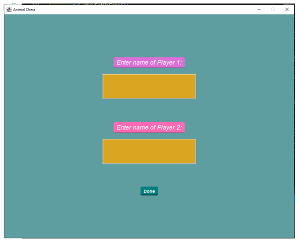
	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the volume is lessen	f = -20	the soundtrack's volume will be lessen	the soundtrack's volume was lessen	Pass

Folder: Controller
Class : GameController
Method:public void startGame()
Method Description: This function starts the game

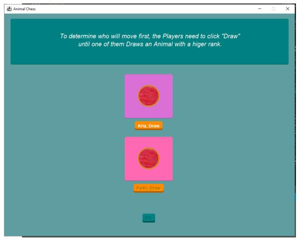
	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the game starts		the main menu will be visible	 Animal Chess Main Menu Start Game How to Play Sound Settings Exit Game	Pass

Folder: Controller

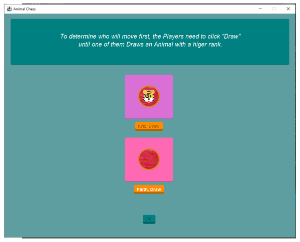
Class : GameController
Method:public void getPlayerNames()
Method Description: This function gets the names of the players during a game

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the players will be asked to enter their names		the set up game will be visible	 Animal Chess Enter name of Player 1: Enter name of Player 2: Done	Pass

Folder: Controller
Class : GameController
Method:public public void drawPlayer1()
Method Description: This function lets player one draw an animal

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player one will be asked to draw an animal		the set up game will be visible	 To determine who will move first, the Players need to click "Draw" until one of them Draws an Animal with a higer rank. 	Pass

Folder: Controller
Class : GameController
Method:public void drawPlayer2()
Method Description: This function lets player two draw an animal

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player two will be asked to draw an animal		the set up game will be visible	 To determine who will move first, the Players need to click "Draw" until one of them Draws an Animal with a higer rank.	Pass

Folder: Controller
Class : GameController
Method:public void checkIfValidDraw()
Method Description: This function checks if a player won the draw by drawing an animal with higher rank

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player one wins the draw		player one will be asked to choose their team's color	player one was asked to choose their team's color	Pass
2	player two wins the draw		player two will be asked to choose their team's color	player two was asked to choose their team's color	Pass
3	player one and player two drew animals of equal rank		players will be asked to draw again	players were asked to draw again	Pass

Folder: Controller
Class : GameController
Method:public void assignColor(boolean bBlue, String strColor)
Method Description: This function assigns a color to the players

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	player one won the draw and chose color BLUE		player one will be BLUE Team and player two will be RED Team	player one was BLUE Team and player two was RED Team	Pass

2	player one won the draw and chose color RED		player one will be RED Team and player two will be BLUE Team	player one was RED Team and player two was BLUE Team	Pass

Folder: Controller

Class : GameController

Method:public void turnSoundOn()

Method Description: This function turns on the soundtrack during the game

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	turnSoundOn() is called		the soundtrack of the game will be unmuted	the soundtrack of the game was unmuted	Pass

Folder: Controller

Class : GameController

Method:public void turnSoundOff()

Method Description: This function turns off the soundtrack during the game

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	turnSoundOff() is called		the soundtrack of the game will be muted	the soundtrack of the game was muted	Pass

Folder: Controller

Class : GameController

Method:public void quitGameReset()

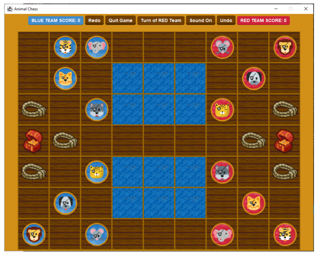
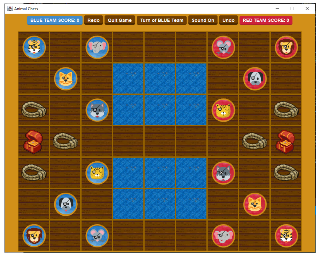
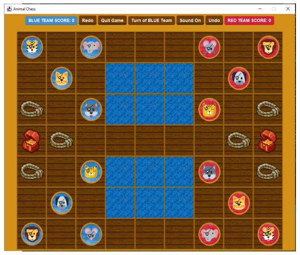
Method Description: This function resets the game and the gui when the Quit Game button is pressed or when a player won the game

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player pressed the “Quit Game” button		the game and gui will reset and the Main Menu will be visible	the game and gui were reset and the Main Menu became visible	Pass

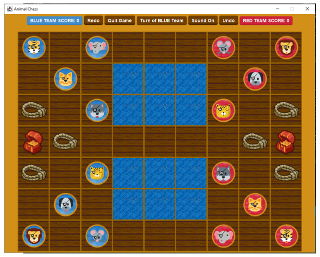
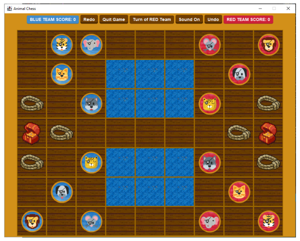
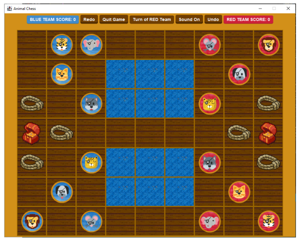
Folder: Controller

Class : GameController

Method:public void undo()
Method Description: This function undoes the changes a player made

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player pressed the "Undo" button	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of RED Team Sound On Undo RED TEAM SCORE: 0	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of BLUE Team Sound On Undo RED TEAM SCORE: 0	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of BLUE Team Sound On Undo RED TEAM SCORE: 0	Pass

Folder: Controller
Class : GameController
Method:public void redo()
Method Description: This function redos the changes a player made

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player pressed the "Redo" button	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of BLUE Team Sound On Undo RED TEAM SCORE: 0	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of RED Team Sound On Undo RED TEAM SCORE: 0	 Animal Chess BLUE TEAM SCORE: 0 Redo Quit Game Turn of RED Team Sound On Undo RED TEAM SCORE: 0	Pass

Folder: Controller
Class : GameController
Method:public void changeSoundTrack(File fileToPlay)
Method Description: This function changes the soundtrack of the game given a File object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	changeSoundTrack() is called		The soundtrack playing will be changed	The soundtrack playing was changed	Pass

Folder: Controller
Class : GameController
Method:public int calculatePostX(MouseEvent e)
Method Description: This function calculates the row pressed during a game given a MouseEvent object

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a player pressed area of the board corresponding to (2,4)		will return 2	returned 2	Pass

Folder: Controller
Class : GameController
Method:public int calculatePostY(MouseEvent e)
Method Description: This function calculates the column pressed during a game given a MouseEvent object

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a player pressed area of the board corresponding to (2,4)		will return 4	returned 4	Pass

Folder: Controller
Class : GameController
Method:public void preDrag(int nInitX, int nInitY)
Method Description: This function assumes a cell has animal, a move is not out of bounds, a piece is own piece, and prepares a piece to be dragged

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a player pressed an area without an animal		nothing will happen	nothing happened	Pass
2	a player pressed a piece that is of their enemy’s		nothing will happen	nothing happened	Pass

3	a player pressed one of their pieces		they will be able to “lift” their piece	they were be able to “lift” their piece	Pass

Folder: Controller
Class : GameController
Method:public void executeDrag(int nPostX, int nPostY)
Method Description: This function executes a drag

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player drags one of their pieces		the player will be able to drag their piece	the player was able to drag their piece	Pass

Folder: Controller
Class : GameController
Method:public void postDrag()
Method Description: This function removes the dragged piece from the drag layer after dragging

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player stops dragging one of their pieces		the piece will be removed from the drag layer	the piece was removed from the drag layer	Pass

Folder: Controller
Class : GameController
Method:public void mousePressed(MouseEvent e)
Method Description: This function implements and overrides the mousePressed(MouseEvent e) and sets the nInitX and nInitY based on the surface pressed

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player pressed one of their pieces		nInitX and nInitY will not be -1	nInitX and nInitY weren’t -1	Pass
2	a player pressed one of their opponent’s pieces		nInitX and nInitY will be -1	nInitX and nInitY were -1	Pass

3	a player pressed an empty tile		nInitX and nInitY will be -1	nInitX and nInitY were -1	Pass

Folder: Controller
Class : GameController
Method:public void mouseReleased(MouseEvent e)
Method Description: This function implements and overrides the mouseReleased(MouseEvent e) and sets the icon of the Pieces object in the BoardPieces object based on the move that a player wants to make

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player released their piece in a valid position		the gui will be updated	the gui was updated	Pass
2	a player released their piece in an invalid position		the gui will not be updated	the gui wasn't updated	Pass

Folder: Controller
Class : GameController
Method:public void mouseDragged(MouseEvent e)
Method Description: This function implements and overrides the mouseDragged(MouseEvent e)

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a player drags their piece		the player will be able to drag their piece that they pressed	the player was able to drag their piece that they pressed	Pass

Class : view.MainMenu
Constructor: public MainMenu()
Constructor: Constructs a new MainMenu object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a MainMenu object is created		a MainMenu Object is created	a MainMenu Object is created	Pass

Class : view.MainMenu
Method: public JPanel getPnlMainMenu()
Method Description: This function returns the pnlMainMenu

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a pnlMainMenu is returned		a pnlMainMenu is returned	a pnlMainMenu is returned	Pass

Class : view.MainMenu
Method: public JButton getBtnHowToPlay()
Method Description: This function returns btnHowToPlay

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnHowToPlay is returned		a btnHowToPlay is returned	a btnHowToPlay is returned	Pass

Class : view.MainMenu
Method: public JButton getBtnSoundSettings()
Method Description: This function returns btnSoundSettings

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnSoundSettings is returned		a btnSoundSettings is returned	a btnSoundSettings is returned	Pass

FOLDER : VIEW | SOME CLASSES UNDER VIEW

Class : view.MainMenu

Method: public JButton getBtnExitGame()

Method Description: This function returns btnExitGame

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnExitGame is returned		a btnExitGame is returned	a btnExitGame is returned	Pass

Class : view.MainMenu

Method: private void setUpMainMenu()

Method Description:This function initializes the attributes of a MainMenu object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the attributes of a MainMenu object have been initialized		the attributes of a MainMenu object have been initialized	the attributes of a MainMenu object have been initialized	Pass

Class : view.HowToPlay

Constructor: public HowToPlay()

Constructor: constructs a new HowToPlay object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a HowToPlay object is created		a HowToPlay object is created	a HowToPlay object is created	Pass

Class : view.HowToPlay

Method: public JPanel getPnlInstructions()

Method Description: This function returns the pnlInstructions

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a pnlInstructions is returned		a pnlInstructions is returned	a pnlInstructions is returned	Pass

Class : view.HowToPlay
Method: private void setUpHowToPlay()
Method Description: This function initializes the attributes of a HowToPlay object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the attributes of a HowToPlay object have been initialized		the attributes of a HowToPlay object have been initialized	the attributes of a HowToPlay object have been initialized	Pass

Class : view.HowToPlay
Method: public void updateHowToPlay(JButton btnBackToMainMenu)
Method Description:This function adds a btnBackToMainMenu to the pnlInstructions of a HowToPlay object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnBackToMainMenu has been added to the pnlInstructions of a HowToPlay object		a btnBackToMainMenu has been added to the pnlInstructions of a HowToPlay object	a btnBackToMainMenu has been added to the pnlInstructions of a HowToPlay object	Pass

Class : view.SoundSettings
Constructor: public SoundSettings()
Constructor Description: Constructs a SoundSettings object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a new SoundSettings object has been initialized		a new SoundSettings object has been initialized	a new SoundSettings object has been initialized	Pass

Class : view.SoundSettings
Method:public JPanel getPnlSoundSettings()
Method Description: This function returns the pnlSoundSettings

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail

1	a pnlSoundSettings is returned		a pnlSoundSettings is returned	a pnlSoundSettings is returned	Pass

Class : view.SoundSettings

Method: public JButton getBtnTrack1()

Method Description: This function returns the btnTrack1

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a btnTrack1 is returned		a btnTrack1 is returned	a btnTrack1 is returned	Pass

Class : view.SoundSettings

Method: public JButton getBtnTrack2()

Method Description: This function returns the btnTrack2

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a btnTrack2 is returned		a btnTrack2 is returned	a btnTrack2 is returned	Pass

Class : view.SoundSettings

Method: public JButton getBtnTrack3()

Method Description: This function returns the btnTrack3

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a btnTrack3 is returned		a btnTrack3 is returned	a btnTrack3 is returned	Pass

Class : view.SoundSettings

Method: public JButton getBtnDefaultTrack()

Method Description: This function returns the btnDefaultTrack

	Test Description	**Input value/ parameters**	**Expected output/ result**	**Actual output/result**	**Pass/Fail**
1	a btnDefaultTrack is returned		a btnDefaultTrack is returned	a btnDefaultTrack is returned	Pass

Class : view.SoundSettings
Method: private void setUpSoundSettings()
Method Description: This function initializes the attributes of a SoundSettings object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the attributes of a SoundSettings object have been initialized		the attributes of a SoundSettings object have been initialized	the attributes of a SoundSettings object have been initialized	Pass

Class : view.SoundSettings
Method: public void updateSoundSettings(JButton btnBackToMainMenu)
Method Description: This function adds a btnBackToMainMenu to the pnlSoundSettings of a SoundSettings object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnBackToMainMenu has been added to the pnlSoundSettings		a btnBackToMainMenu has been added to the pnlSoundSettings	a btnBackToMainMenu has been added to the pnlSoundSettings	Pass

Class : view.SetUpGame
Constructor: public SetUpGame()
Constructor Description: Constructs a new SetUpGame object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a SetUpGame object has been constructed		a SetUpGame object has been constructed	a SetUpGame object has been constructed	Pass

Class : view.SetUpGame
Method: public JPanel getPnlSetUpGame()
Method Description: This function returns the pnlSetUpGame

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a pnlSetUpGame is returned		a pnlSetUpGame is returned	a pnlSetUpGame is returned	Pass

Class : view.SetUpGame
Method: public JTextField getTxtfldPlayer1()
Method Description: This function returns the txtfldPlayer1

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a the txtfldPlayer1 is returned		a the txtfldPlayer1 is returned	a the txtfldPlayer1 is returned	Pass

Class : view.SetUpGame
Method: public JTextField getTxtfldPlayer2()
Method Description: This function returns the txtfldPlayer2

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a the txtfldPlayer2 is returned		a the txtfldPlayer2 is returned	a the txtfldPlayer2 is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnDone()
Method Description: This function returns the btnDone

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnDone is returned		a btnDone is returned	a btnDone is returned	Pass

Class : view.SetUpGame
Method: public JPanel getPnlDrawAnimal()
Method Description: This function returns the pnlDrawAnimal

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a pnlDrawAnimal is returned		a pnlDrawAnimal is returned	a pnlDrawAnimal is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnDrawPlayer1()
Method Description: This function returns the btnDrawPlayer1

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnDrawPlayer1 is returned		a btnDrawPlayer1 is returned	a btnDrawPlayer1 is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnDrawPlayer2()
Method Description: This function returns the btnDrawPlayer2

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnDrawPlayer2 is returned		a btnDrawPlayer2 is returned	a btnDrawPlayer2 is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnDoneDraw()
Method Description: This function returns the btnDrawDone

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnDrawDone is returned		a btnDrawDone is returned	a btnDrawDone is returned	Pass

Class : view.SetUpGame
Method: public JPanel getPnlChooseColor()
Method Description: This function returns the pnlChooseColor

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a pnlChooseColor is returned		a pnlChooseColor is returned	a pnlChooseColor is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnBlueTeam()
Method Description: This function returns the btnBlueTeam

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnBlueTeam is returned		a btnBlueTeam is returned	a btnBlueTeam is returned	Pass

Class : view.SetUpGame
Method: public JButton getBtnRedTeam()
Method Description: This function returns the btnRedTeam

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a btnRedTeam is returned		a btnRedTeam is returned	a btnRedTeam is returned	Pass

Class : view.SetUpGame
Method: private void setupSetUpGame()
Method Description: This function initializes the attributes of a SetUpGame object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the attributes of a SetUpGame object has been initialized		the attributes of a SetUpGame object has been initialized	the attributes of a SetUpGame object has been initialized	Pass

Class : view.SetUpGame
Method: public void updatePnlDrawAnimal(String strName1, String strName2, String strAnimal1, String strAnimal2)
Method Description: This function sets the btnDrawPLayer1, btnDrawPlayer2, lblDrawAnimal1, and lblDrawAnimal2

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	btnDrawPLayer1, btnDrawPlayer2, lblDrawAnimal1, and lblDrawAnimal2 have been updated		btnDrawPLayer1, btnDrawPlayer2, lblDrawAnimal1, and lblDrawAnimal2 have been updated	btnDrawPLayer1, btnDrawPlayer2, lblDrawAnimal1, and lblDrawAnimal2 have been updated	Pass

Class : view.SetUpGame
Method: public void updateLbl1(String strAnimal1)

Method Description: This function updates the icon of lblDrawAnimal1

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the icon of the lblDrawAnimal1 has been updated		the icon of the lblDrawAnimal1 has been updated	the icon of the lblDrawAnimal1 has been updated	Pass

Class : view.SetUpGame
Method: public void updateLbl2(String strAnimal2)
Method Description: This function updates the icon of lblDrawAnimal2

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the icon of the lblDrawAnimal2 has been updated		the icon of the lblDrawAnimal2 has been updated	the icon of the lblDrawAnimal2 has been updated	Pass

Class : view.SetUpGame
Method: public void updatePnlChooseColor(String strDrawWinner)
Method Description: This function sets the text of lblPlayerToChooseColor

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	the text of the lblPlayerToChooseColor has been updated		the text of the lblPlayerToChooseColor has been updated	the text of the lblPlayerToChooseColor has been updated	Pass

Class : view.Board
Constructor: public Board()
Constructor Description: Constructs a new Board object

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	a new Board object has been created		a new Board object has been created	a new Board object has been created	Pass

Class : view.Board
Method: public JPanel getPnlBoard()
Method Description: This function returns the pnlBoard

	Test Description	Input value/ parameters	Expected output/ result	Actual output/result	Pass/Fail
1	returns a pnlBoard		returns a pnlBoard	returns a pnlBoard	Pass