# Coursework 2 – Tic-Tac-To: Markov Decision Processes & Reinforcement Learning (worth 25% of your final mark)

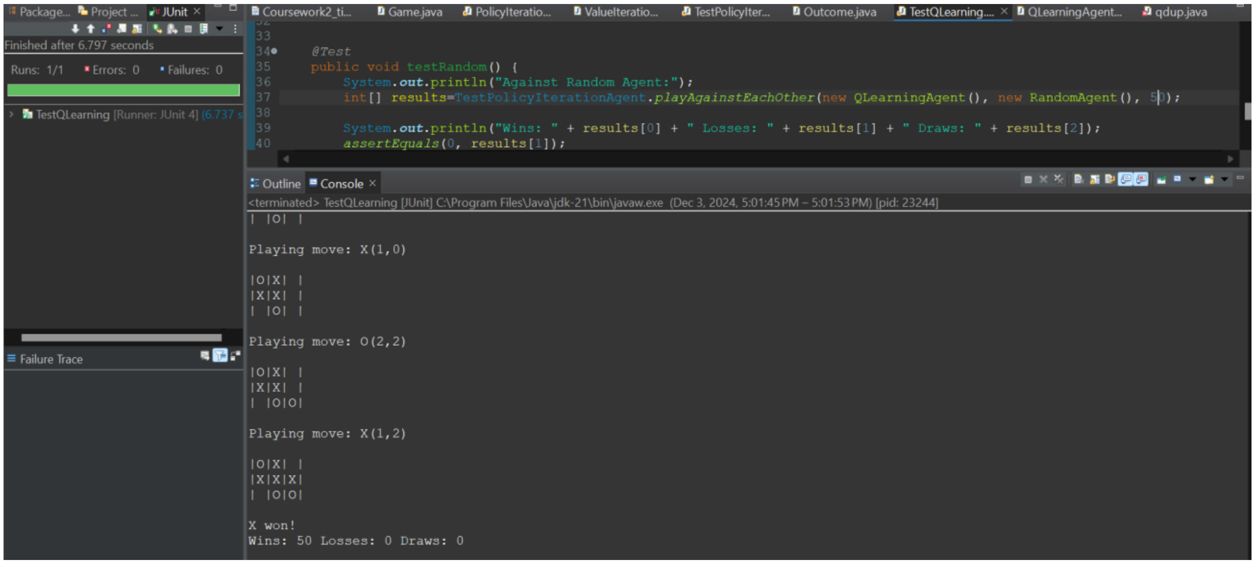
***Question 6 (1 point):*** Like the previous questions, test your Q-Learning Agent against each of the provided agents 50 times and report on the results - how many games they won, lost & drew. The other agents are: *random*, *aggressive*, *defensive.*

**Ans**:

- Q learning agent uses epilson greedy function to exploit and explore.
- Train() : goes through a number of episodes using an epsilon-greedy strategy. During each episode, the agent updates Q-values based on rewards and resets the environment after reaching a terminal state
- extractPolicy(): maps each game to its best move on q value in q table.

| | WINS | LOSS | DRAWS |
|---|---|---|---|
| **RANDOM** | 50 | 0 | 0 |
| **AGRESSIVE** | 50 | 0 | 0 |
| **DEFENSIVE** | 44 | 0 | 6 |

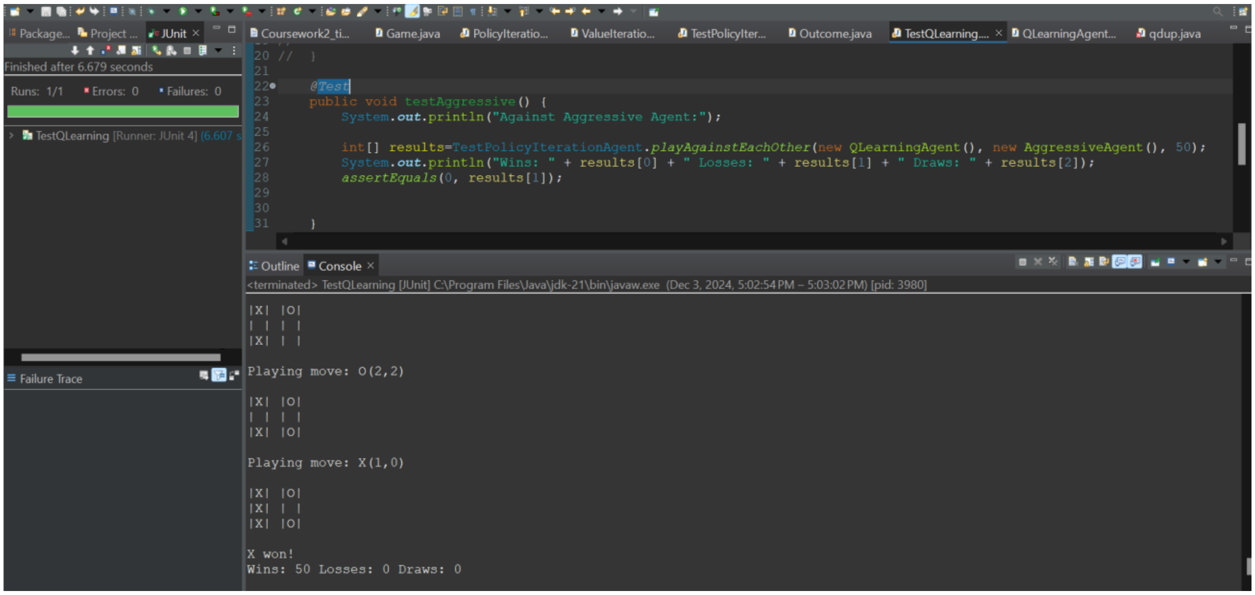
Random:



```
    @Test
    public void testRandom() {
        System.out.println("Against Random Agent:");
        int[] results=TestPolicyIterationAgent.playAgainstEachOther(new QLearningAgent(), new RandomAgent(), 50);

        System.out.println("Wins: " + results[0] + " Losses: " + results[1] + " Draws: " + results[2]);
        assertEquals(0, results[1]);
```

```
<terminated> TestQLearning [JUnit] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Dec 3, 2024, 5:01:45 PM – 5:01:53 PM) [pid: 23244]
| |O| |

Playing move: X(1,0)

|O|X| |
|X|X| |
| |O| |

Playing move: O(2,2)

|O|X| |
|X|X| |
| |O|O|

Playing move: X(1,2)

|O|X| |
|X|X|X|
| |O|O|

X won!
Wins: 50 Losses: 0 Draws: 0
```

Aggressive:



Finished after 6.679 seconds

Runs: 1/1 Errors: 0 Failures: 0

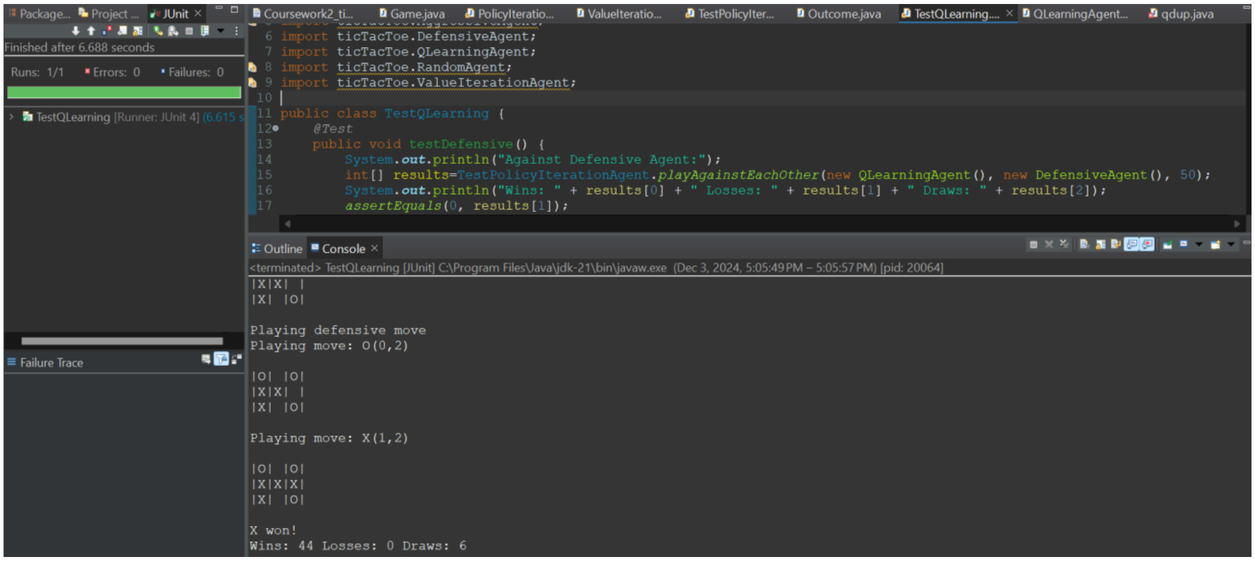
TestQLearning [Runner: JUnit 4] (6.607 s

```
//   }

    @Test
    public void testAggressive() {
        System.out.println("Against Aggressive Agent:");

        int[] results=TestPolicyIterationAgent.playAgainstEachOther(new QLearningAgent(), new AggressiveAgent(), 50);
        System.out.println("Wins: " + results[0] + " Losses: " + results[1] + " Draws: " + results[2]);
        assertEquals(0, results[1]);


    }
```

<terminated> TestQLearning [JUnit] C:\Program Files\Java\jdk-21\bin\javaw.exe (Dec 3, 2024, 5:02:54 PM – 5:03:02 PM) [pid: 3980]

```
|X|  |O|
| | | |
|X| | |

Playing move: O(2,2)

|X|  |O|
| | | |
|X|  |O|

Playing move: X(1,0)

|X|  |O|
|X| | |
|X|  |O|

X won!
Wins: 50 Losses: 0 Draws: 0
```

Defensive:



Finished after 6.688 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestQLearning [Runner: JUnit 4] (6.615 s

```
import ticTacToe.DefensiveAgent;
import ticTacToe.QLearningAgent;
import ticTacToe.RandomAgent;
import ticTacToe.ValueIterationAgent;

public class TestQLearning {
    @Test
    public void testDefensive() {
        System.out.println("Against Defensive Agent:");
        int[] results=TestPolicyIterationAgent.playAgainstEachOther(new QLearningAgent(), new DefensiveAgent(), 50);
        System.out.println("Wins: " + results[0] + " Losses: " + results[1] + " Draws: " + results[2]);
        assertEquals(0, results[1]);
```

<terminated> TestQLearning [JUnit] C:\Program Files\Java\jdk-21\bin\javaw.exe (Dec 3, 2024, 5:05:49 PM – 5:05:57 PM) [pid: 20064]

```
|X|X|  |
|X|  |O|

Playing defensive move
Playing move: O(0,2)

|O|  |O|
|X|X|  |
|X|  |O|

Playing move: X(1,2)

|O|  |O|
|X|X|X|
|X|  |O|

X won!
Wins: 44 Losses: 0 Draws: 6
```