

Program Structures and Algorithms

Spring 2024

Team Members: Siddharth Dumbre, Gaurav Gunjal

GITHUB LINK: https://github.com/gunjalga/INFO6205_MCTS

Task: Tic Tac Toe

Screenshots:

The screenshot displays an IDE with the following components:

- Editor:** Shows the `PositionTest.java` file. The code includes package declarations, imports, and a `testMove_2()` method that tests a move on a 3x3 grid.
- Run Console:** Displays the execution results of the tests. It shows that all 18 tests passed successfully within a total time of 24 ms.
- Test Results Table:** A table listing the individual tests, their durations, and the expected outcomes.

Test Name	Duration	Expected Outcome
testProjectDiag	4 ms	1,-1,0
testMove0	4 ms	1,0,-1
testMove1	0 ms	1,0,-1
testMoves	1 ms	1,-1,0
testParseCell	7 ms	1,-1,-1
testFull	1 ms	-1,0,-1
testToString	0 ms	-1,-1,1
testThreeInARow	1 ms	1,-1,-1
testReflect	0 ms	-1,0,-1
testWinner0	1 ms	-1,-1,1

```

14  *
15  */
16  @Gaurav Popat Gunjal
17  @Test
18  public void runGame() {
19      long seed = 0L;
20      TicTacToe target = new TicTacToe(seed); // games run here will all be deterministic.
21      State<TicTacToe> state = target.runGame();
22      Optional<Integer> winner = state.winner();
23      if (winner.isPresent()) assertEquals(Integer.valueOf(TicTacToe.X), winner.get());
24      else fail("no winner");
25  }
26

```

Run **TicTacToeTest** x

✓ **TicTacToeTest** (edu.neu.coe.5 ms) ✓ Tests passed: 1 of 1 test - 5 ms

✓ runGame 5 ms

```

/Users/siddharth/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java ...
-1,-1,-1
-1,-1,-1
-1,-1,-1
Process finished with exit code 0

```

FO6205_MCTS > src > test > java > edu > neu > coe > info6205 > mcts > tictactoe > TicTacToeTest 26:4 LF UTF-8 4 spaces

```

1  package edu.neu.coe.info6205.mcts.tictactoe;
2
3  > import ...
9
10 @Gaurav Popat Gunjal *
11 public class TicTacToeNodeTest {
12     @Gaurav Popat Gunjal *
13     @Test
14     public void winsAndPlayouts() {
15         TicTacToe.TicTacToeState state = new TicTacToe().new TicTacToeState(Position.parsePosition( grid: "X . 0\nX 0 .\nX . 0", TicTacToe.X));
16         TicTacToeNode node = new TicTacToeNode(state);
17         assertTrue(node.isLeaf());
18         assertEquals( expected: 1, node.wins());
19         assertEquals( expected: 1, node.playouts());
20     }
21 }

```

Run **TicTacToeNodeTest** x

✓ **TicTacToeNodeTest** (edu.r.6 ms) ✓ Tests passed: 8 of 8 tests - 6 ms

✓ addChild 2 ms ✓ isLeaf 1 ms ✓ state 3 ms ✓ white 0 ms ✓ backPropagate 0 ms ✓ simulateRandom 0 ms ✓ children 0 ms ✓ winsAndPlayouts 0 ms

```

/Users/siddharth/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java ...
-1,-1,-1
-1,-1,-1
-1,-1,-1
-1,-1,-1
-1,-1,-1
-1,-1,-1
1,-1,0
1,0,-1
1,-1,0

```

FO6205_MCTS > src > test > java > edu > neu > coe > info6205 > mcts > tictactoe > TicTacToeNodeTest > winsAndPlayouts 16:35 LF UTF-8 4 spaces

Conclusion:

The move method in the Position class takes the player and the coordinates of the move for the player. It fills a new position and updates the existing position.

The moves method takes the player and fills it with all the possible moves from that position.

The threeInARow method checks if a certain player has won by checking all the rows, columns, and diagonals. If the 3 values are the same then it returns true, else false.