

INFO 6205 Spring 2022 Project

Menace



- **Introduction**

- **Aim: To train the MENANCE to win against the human**
- **Approach:**

If a human plays with all strategies it is bound that the human will win approximately every time. Hence, we have tried to implement the human strategy with random moves. This will help the menace to train to either win or lose.

The menace trains itself by playing with the random player. There are 304 matchboxes assuming containing beads. When the game begins the menace checks the symmetry between the match boxes and the current game state and choose the move to play. It plays until it reaches a conclusion state i.e. win or lose or draw.

- **Program**

- **Data Structures & classes**

1. HashMap (to store the states of the various matchboxes)
2. Array (to store the various wins and loss state in the game)

- **Algorithm**

1. Start
2. Menace has a computer and human as players playing on 3*3 matrix board. Where the player is chosen randomly and take turns to play the game.
3. *int numberOfPlays = TRAINING_ROUNDS*
Human plays manually whereas the computer is trained with a random player who uses random moves with probability p!.
4. *if(game.getState() == GameState.PLAYING)*
{
If(game.getValueAt(j) == CellValue.EMPTY)
{
MenaceTicTacToeGame newgame = new
MenaceTicTacToeGame(game, j);
}
}
Menace choses to play random beads between the matchboxes until a winning move is acquired.
5. The symmetry is checked several times until the game is transformed to the current game state the players move are based upon the transformation.
6. The above steps are repeated unless a conclusion arrives .

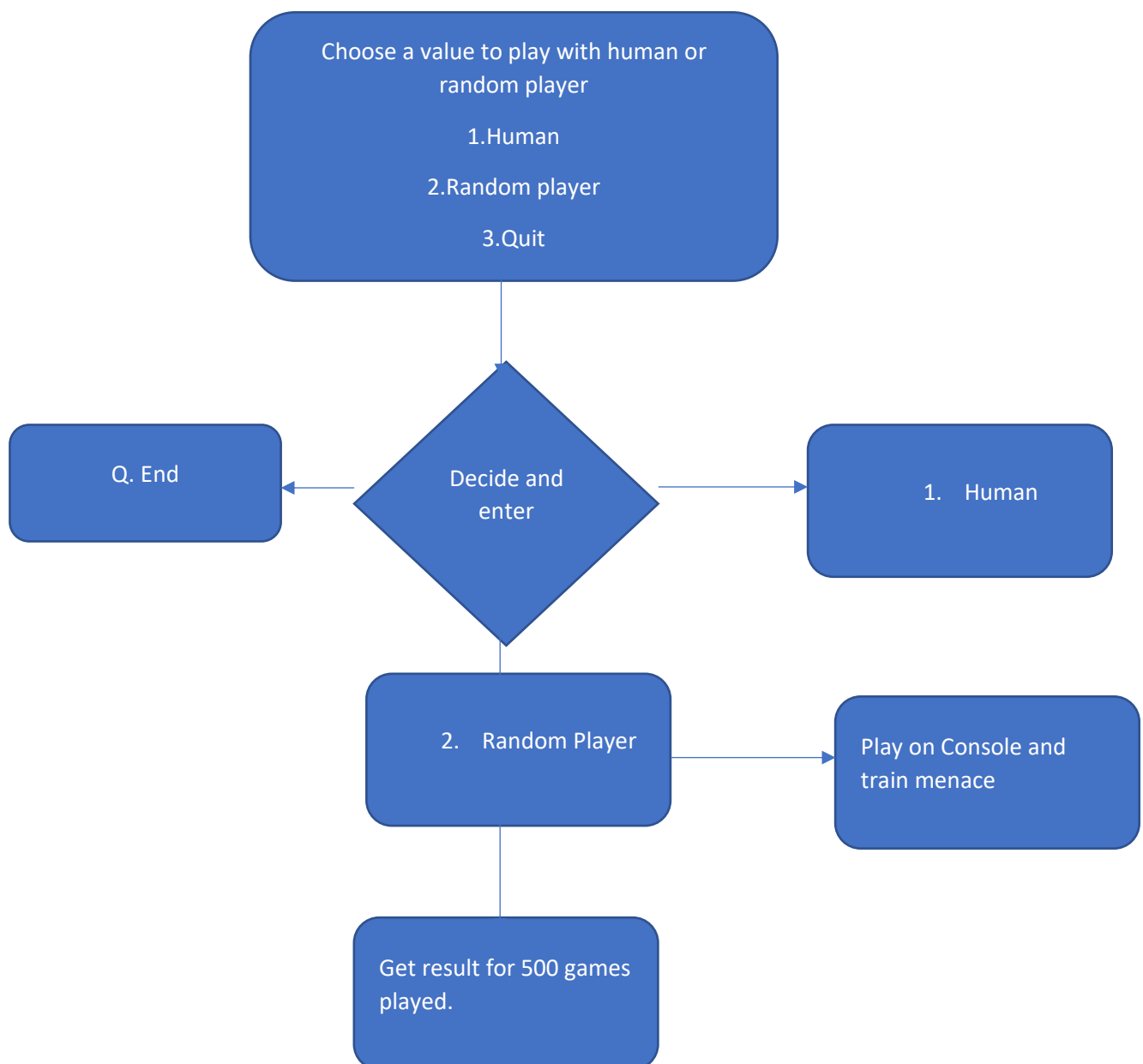
- **Invariants**

The invariant is a condition or a relation which is always true.

1. The relationship between of X's and O's in the board will not change even with the variation of their count.
2. The winning condition is if either X or O are diagonally present.
3. The wining condition is if either X or O come in a row or column altogether.

- **Flow Charts (inc. UI Flow)**

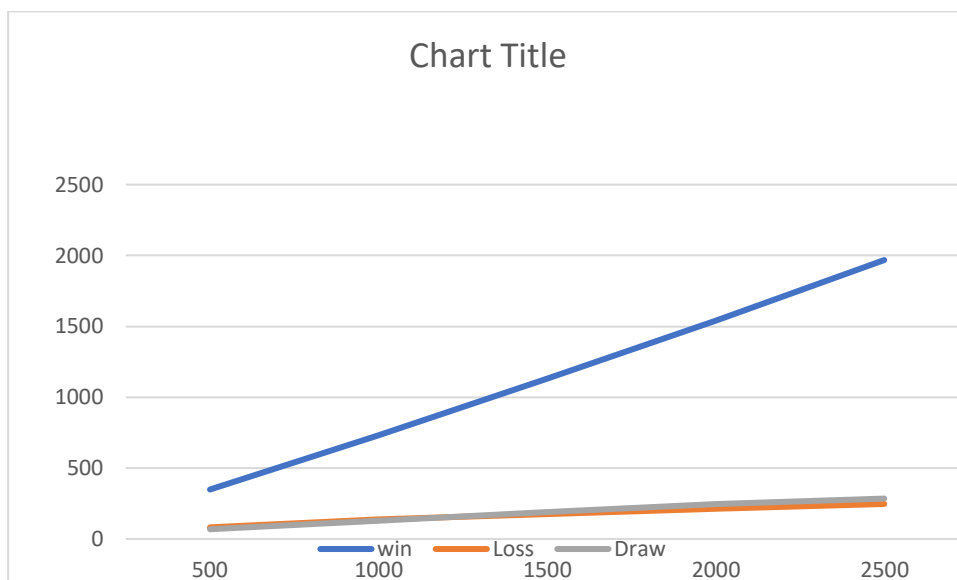
Following are the steps to follow.



- **Observations & Graphical Analysis**

MENACE works similar to neural network. It behaves random at the beginning and eventually is optimized after a few games. It stores the moves which led to win and trains itself. Depending on the board state it will win, lose or draw and reward or punish itself by adding the number of beads accordingly.

Number of games	Win	Loss	Draw
500	349	82	69
1000	731	138	131
1500	1132	176	192
2000	1538	214	248
2500	1968	247	285



- **Results & Mathematical Analysis**

Player 1: This player has won 349 games, lost 82 games and 69 were draws. Over the last 50 games, this player has won 38 games, lost 4 games and 8 were draws.

The probability for Menace to win for 500 games against the random human is .698, loss is 0.164 and draw is 0.138.

Player 1: This player has won 731 games, lost 138 games and 131 were draws. Over the last 50 games, this player has won 34 games, lost 7 games and 9 were draws.

The probability for Menace to win for 1000 games against the random human is 0.731, loss is 0.138 and draw is 0.131.

Player 1: This player has won 1150 games, lost 176 games and 174 were draws. Over the last 50 games, this player has won 39 games, lost 3 games and 8 were draws.

The probability for Menace to win for 1500 games against the random human is 0.766, loss is 0.1176 and draw is 0.116.

Player 1: This player has won 1538 games, lost 214 games and 248 were draws. Over the last 50 games, this player has won 40 games, lost 3 games and 7 were draws.

The probability for Menace to win for 2000 games against the random human is 0.769, loss is 0.107 and draw is 0.124.

Player 1: This player has won 1968 games, lost 247 games and 285 were draws. Over the last 50 games, this player has won 42 games, lost 5 games and 3 were draws.

The probability for Menace to win for 2500 games against the random human is 0.787, loss is 0.098 and draw is 0.114.

With the above mathematical results, we observe that the menace is getting trained and is gradually winning.



```
Run: TicTacToe
(5) Delete (both) Menace training sets
(Q)uit

player 1: This player has won 1968 games, lost 247 games and 285 were draws. Over the last 50 games, this player has won 42 games, lost 5 games and 3 were draws.
player 2: This player has won 247 games, lost 1968 games and 285 were draws. Over the last 50 games, this player has won 5 games, lost 42 games and 3 were draws.
(1) Human to play menace
(2) Train Menace against perfect player
(3) Train Menace against random player
(4) Train Menace against another menace
(5) Delete (both) Menace training sets
(Q)uit
```

Player 1: This player has won 1968 games, lost 247 games and 285 were draws. Over the last 50 games, this player has won 42 games, lost 5 games and 3 were draws.

Player 2: This player has won 247 games, lost 1968 games and 285 were draws. Over the last 50 games, this player has won 5 games, lost 42 games and 3 were draws.

Player 1(Menace): This player has won 4609 games, lost 480 games and 411 were draws. Over the last 50 games, this player has won 41 games, lost 3 games and 6 were draws.

Player 2(RandomPlayer): This player has won 480 games, lost 4609 games and 411 were draws. Over the last 50 games, this player has won 3 games, lost 41 games and 6 were draws.

1. At the start of the game, the number of beads in each matchbox is displayed as alpha.

- a) In the first Matchbox, there are 8 beads (which is the starting representation of the game).
 - i. Out of a total of nine moves, there have been zero.
 - ii. Menace is the first to act.
- b) In the next 12 Matchboxes, 4 beads (which is where Human and Menace both have played once)
 - i. There are a total of two moves out of nine.
 - ii. Menace makes the third move.
- c) in the next 108 Matchboxes, 2 beads (which is where Human and Menace both have played twice)
 - i. Total of 4 out of 9 moves
 - ii. Menace makes his 5th move
- d) 1 bead in each of the remaining 183 Matchboxes (which is where Human and Menace both have played thrice)

- i. There are a total of 6 moves out of a total of 9
 - ii. Menace makes the 7th move.
- e) There are no more Matchboxes or beads to represent, so the last (9th) and final move will determine whether the game is won by MENACE or drawn.

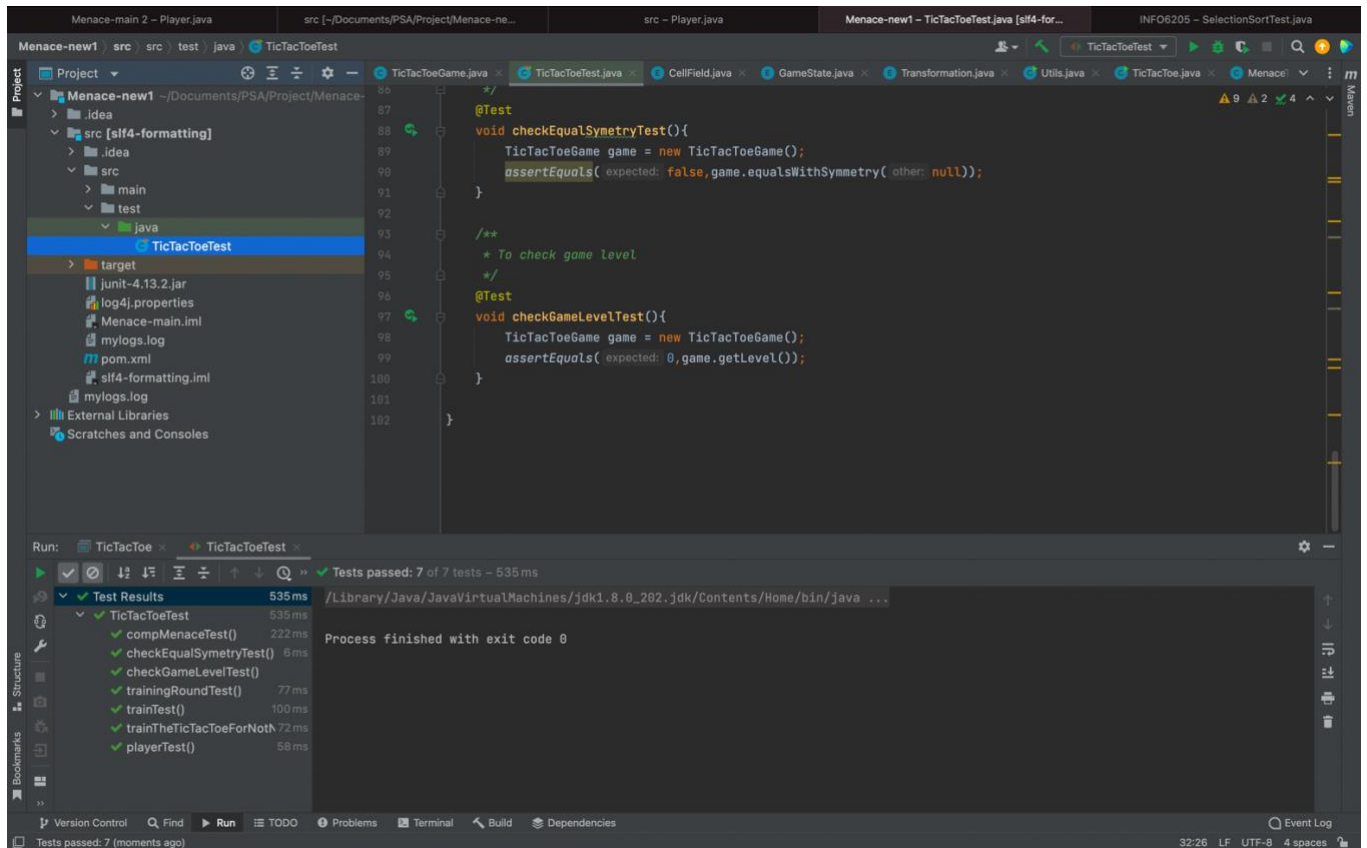
2. When the MENACE wins the beta phase of the match, he is rewarded with beads. Note: Three beads are given to each of the 304 matchboxes that contributed to the game's victory.

3. When MENACE loses a match, the gamma phase, the beads are reduced to penalize the MENACE. Note: Each of the 304 matchboxes that contributed to the game's loss is penalized by losing one bead.

4. In the delta phase, when the game is tied (neither MENACE nor Human wins), a bead is awarded. Note: Each of the 304 matchboxes that contributed to the game's drawing is rewarded with one bead.

The screenshot shows an IDE with a project named 'Menace-new1'. The 'src' directory contains several files, including 'mylogs.log'. The log file is open, displaying a series of log entries for a TicTacToe game. Each entry includes a timestamp (27-04-2022 21:01:36), a log level (INFO), and a message from 'TicTacToe:27'. The messages describe the state of the game board at various points, using a 4x4 grid representation where 'X' and 'O' denote player positions and 'EMPTY' denotes empty cells. The log entries show a sequence of moves and state changes, with the board state being updated after each move. The IDE interface also shows a 'Project' view on the left with the file structure of 'Menace-new1', including 'src', 'target', and 'log4j.properties'. The bottom status bar indicates 'BUILD completed successfully with 3 warnings in 2 sec. 698 ms (40 minutes ago)'.

- **Testcases**



- **Conclusion**

In our project, we have implemented a machine capable of learning to be a better player of tic tac toe. Menace is built with 304 match boxes. Each match box represents a specific board layout. In real game, whenever the menace is playing it will try to check for these layout boxes and choose the similar layout present. With the game end it will store the result in the hash table and remember move every time the same layout occurs with any flip of the board such as horizontal, vertical, rotate or symmetrical. With every game, the menace rewards itself with +3 beads if it wins and -1 beads if it loses. Hence, the menace trains itself.

In our project we tried to play 500 games at each run and train the menace. We observed that, the probability if menace wining increase with every training.

- **References**

1. [Christmas Lectures 2019: How to Bend the Rules - Hannah Fry](#)
2. [Wikipedia](#)
3. [How 300 Matchboxes Learned to Play Tic-Tac-Toe Using MENACE](#)

Points awarded:

On-time: 50

Report (see above): 20

Scope (training and logs): 4

Code quality: 10

Unit tests: 8

Repository: 5

Teamwork: 3