**1.Introduction**

**Aim:** To Implement “The Menace” by storing each and every state of the game as key and value as a list of favourable moves in a hash table and train it effectively to make optimal moves.

**Approach:** We will train the menace by making it to play with humans. Every game experience will be captured by the menace machine. With every kind of move made by the menace machine, a reward is granted which will train the machine to make best sensible moves in future games.

**2. Program**

**Data Structures & classes:**

1.HashMap<String, StringScore>:

Every record will store:

1.Key: String

To store the state of the game.

2.Value: Class StringScore

To store the counter move and score.

2.HashMap<String, String>:

Every record will store:

1.Key: String

To store the state of the game before next move.

2.Value: String

To store the state of the game after next move.

3.ArrayList<String>: To store set of moves in a game.

**Classes**:

1. TicTacToe: Entry class having main method to start and initialize the game.

2. StringScore: To store moves and counter moves.

3. ImproveMove : Class logics contains methods to improve the counter moves while training the menace.

**ALGORITHM**

Step I] Give options  
1.play against menace  
2 Train the menace  
3.Exit

Step II] Initialization

1. Initialize a blank 3x3 size board.  
2. Declare a character variable which can be either 'x' or 'o' depending on the players turn.  
3. Initialize the hash map to store moves & counter moves of the game.  
4. Initialize the value of alpha (initial state), beta (win state), gamma (tie state), delta(loss).

Step III a] If user selects option 1: Play against menace.  
1. Ask for player's name.  
3. Ask the player to make a move (ask for row number & column number).  
4. Modify the board and display.  
5. Now it's menace's turn, Check the state of the board and search for counter move in HashMap. While selecting counter move, select the move with best score. If move not found in hashmap make random move.  
6. Display the board. Store the move & counter move in array list.  
7. Check if any player has won/loss or the board is full.  
8. Else Go to point 3.  
9. When the game ends copy the moves from array list to hash map. Give every move a score depending on win/loss/tie.

Step III b] If user selects option 2: Train the menace

1. initialize a dummy player with turn ‘x’ & computer's turn will be ‘o’.

2. Dummy player will make a move on board using following logic  
- If there are 2 x's in same row/column/ diagonal make move to win.  
- If opponent has 2'o in same row/column/ diagonal make move to avoid loss  
- else make a random move on board.

3. Computers will make a random move on board where there is blank space.

4. After every move, check if there are 3 'x' or 3'o' in same row/column/diagonal.  
-if there are 3 ‘x’ in row/column/diagonal player with x wins  
-if there are 3 ‘o’ in row/column/diagonal player with o wins  
- else if board is full, it’s a tie  
5. Go to step 2 if board is not full.  
6. Display the move. Store the move & counter move in array list.  
7. At the end of the game, if any player wins/loss/ time, store the moves from array list to hash map and give them a score(alpha/beta/gamma/delta) .

Step IV] If user selects option 3. Exit.

**Invariants**:

1.Player can mark only in the blank space.

2.It’s a 3x3 matrix board and players can make the move within 3\*3 empty spaces

3.The player can choose either X or O at the beginning of the game and must stick to it throughout the game.

4.It’s a two-player game with alternate turns.

**3.Flow Chart:**  
Diagram

Description automatically generated

**4. Observation & Graphical Analysis:**

Please refer the attachment file.

**5.Results and Mathematical Analysis :**

Graphical user interface, text, application

Description automatically generated with medium confidence

If we don’t train the menace machine, it loses all the games.

If we train the menace machine for 50 -100 times, it won only one game & it lost most of the games.

If we train the menace machine for 200 -300 times, it made better counter moves & won 5 times & tied 9 times.

If we train the menace machine for 400 -500 times, it made better counter moves & won 6 times & tied 8 times.

**6.Test Cases**:

Please refer the attachment file.

**7.Conclusion:**

Initially, Menace machine loses all the games against the opponent. But as we train the menace machine, it learns to make counter moves and manages to win or tie.