TIC TAC TOE

As a simple AI Game

END TERM REPORT

By

Section : K18RD

Chitrothu Bhavya Sree
Roll no : 30

Sparsh Maheshwari
Roll no : 39

Gunda Lalitha Sai Tharun
Roll no : 41
&
Jahnavi Prasad Srirampurapu
Roll no : 42

Department of Intelligent Systems

School of Computer Science and Engineering

Lovely Professional University, Jalandhar

04-2020

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We ever that if any part of the report is found to be copied, we are shall take full responsibility for it.

Name: Bhavya Sree
Reg no: 11808412
Roll no: 30

Name: Sparsh Maheshwari
Reg no: 11813950
Roll no: 39

Name: Sai Tarun
Reg no:11802303
Roll no: 41

Name: Jahnavi Prasad
Reg no: 11803179
Roll no: 42

Place: Lovely professional university.

Date: 08-april-2020.

TABLE OF CONTENTS

Quick Introduction to the project

The project allotted to our team was described as “Develop a simple AI game”. So, after careful discussion and consideration among ourselves we decided to make a AI version of the classic game Tic-Tac-Toe.

About Tic Tac Toe

Tic Tac Toe (also popularly known as “XO”) is a two player game, where each of the player picks a character either “X” or ”O”. the players have a 3*3 matrix before them. One of the players will start the game by drawing their character in a slot of the matrix.

Players take turns placing their X or O. If a player gets three of their marks on the board in a row, column or one of the two diagonals, they win. When the board fills up with neither player winning, the game ends in a draw.

There are multiple variants of the game available with the matrix configuration of n*n with “n” being the number of rows or columns.

Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees.

The game can be generalized to an m,n,k-game in which two players alternate placing stones of their own color on an m×n board, with the goal of getting k of their own color in a row. Tic-tac-toe is the (3,3,3)-game.

Frank Harary's generalized tic-tac-toe is an even broader generalization of tic-tac-toe. It can also be generalized as a n^d game. Tic-tac-toe is the game where n equals 3 and d equals 2. If played optimally by both players, the game always ends in a draw, making tic-tac-toe a futile game.

Implementation

Background :

The childhood memories of the game, experience of building several programs in python in INT213 combined with concepts of artificial intelligence and machine learning in INT404 have served as the much required background for building this project.

Motivation :

The aspiration for implementing new learnt concepts in INT404 was the prime motivation coupled with the opportunity to turn a classic game into a artificially intelligent tool for CA3 is the motivation for building this project.

Outcome :

The outcome of the project is a single python module with artificially intelligent code enough to help a computer think and play the game of Tic Tac Toe counter to a human being. The python module can be run on any python compiler by pre-importing pygame module.

Work division among the team

Chitrothu Bhavya Sree
Roll no : 30
Work :

1. Information gathering
2. Research on the game history
3. Report writing
4. Testing

Sparsh Maheshwari
Roll no : 39
Work :

1. Coding the game
2. Debugging
3. Data collection
4. Algorithm preparation

Gunda Lalitha Sai Tharun
Roll no : 41
Work :

1. Testing
2. Data collection

Jahnavi Prasad Srirampurapu
Roll no : 42
Work :

1. Coding
2. Debugging
3. Researching source material
4. Algorithm preparation and implementation
5. Report writing

Algorithm

We used minimax algorithm for this project.

```
def score(game, depth)
    if game.win?(@player)
        return 10 - depth
    elsif game.win?(@opponent)
        return depth - 10
    else
        return 0
    end
end

def minimax(game, depth)
    return score(game) if game.over?
    depth += 1
    scores = [] # an array of scores
    moves = []  # an array of moves

    # Populate the scores array, recursing as needed
    game.get_available_moves.each do |move|
        possible_game = game.get_new_state(move)
        scores.push minimax(possible_game, depth)
        moves.push move
    end

    # Do the min or the max calculation
    if game.active_turn == @player
        # This is the max calculation
        max_score_index = scores.each_with_index.max[1]
        @choice = moves[max_score_index]
        return scores[max_score_index]
    else
        # This is the min calculation
        min_score_index = scores.each_with_index.min[1]
        @choice = moves[min_score_index]
        return scores[min_score_index]
    end
end
```

References

https://playtictactoe.org/

https://www.anaconda.com/

https://www.pygame.org/

https://en.wikipedia.org/wiki/Tic-tac-toe

https://www.neverstopbuilding.com/blog/minimax

BONAFIDE CERTIFICATE

It is Certified that this project report titled “Tic Tac Toe : As a simple AI game” for the project named “Create a simple AI game” is the bonafide work of “Jahnavi Prasad Srirampurapu, Sparsh Maheshwari, Bhavya Sree and Sai Tarun” who carried out the project work under my supervision.

(Signature of the supervisor)

(Name of the supervisor)

(Academic designation)

(ID of the supervisor)

(Department of the supervisor)