Name: Fuzail Hamid Banday

Title: Tic-Tac-Toe, testing an algorithm for smart AI

Nottingham ID: 20408989

Course: Mechatronics Engineering

Module: EEEE1042 – Introduction to Software Engineering and Programming.

Introduction

I first had to make an algorithm for the Smart AI as this was going to be the basis for my data collection and the whole experiment. The Smart AI used the following logic to place its next move.

Let's say 'X' is the Smart AI and 'O' is Random AI or a Human. First off, the Smart AI goes through to every open spot and places an 'X' there, one by one, and then check if any of these results in a win (Note that this happens on a separate temporary board which is created by copying the original board). In other words, it essentially does the same thing as checking if there are two 'X's in a line, it then places an 'X' on the third spot to win the game. However, if no two 'X's line up, it then checks if there are any two 'O's that line-up, if there are, it places an X in the 3rd spot to essentially block the opponent from winning. If none of these criteria is met, it falls back to the Random AI, which performs a random, legal move. The random AI uses the `rand` function in C whose seed is randomly generated based on the current time using `srand(clock())`.

There were three scenarios, each was tested 100000 times, and for each run, the game would randomly choose which player starts (more on that later). The three scenarios were as follows:

1. Smart AI vs Smart AI.
2. Smart AI vs Random AI.
3. Random AI vs Random AI.

Hypothesis

If you have played tic-tac-toe before, you might know that we often end up in situations where the opponent has 2 possible ways of winning, so even if you try to block one, he will still win the game as in *Figure 1*.

In a situation like this, the player that starts the game (X in this case) is at an advantage. As I mentioned earlier, during my testing, instead of alternating between 'X' starting the game and 'O' starting the game it would randomly choose which starts the game and the random function would mean one of either X or O could get to start the game more times than the other (because the random function is not exactly 50-50). This then could translate to one of them winning the game more times even though the algorithm is the same in Smart AI vs Smart AI or Random AI vs Random AI.

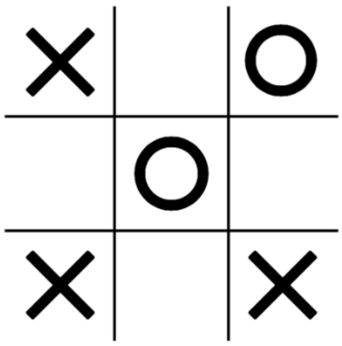


Figure 1 – For example it's O's turn now, and O is the smart AI. X will still win no matter where O is placed

Data collection and results

For each scenario, the game was run 100000 times. The results are as follows:

1) Smart AI vs Smart AI:

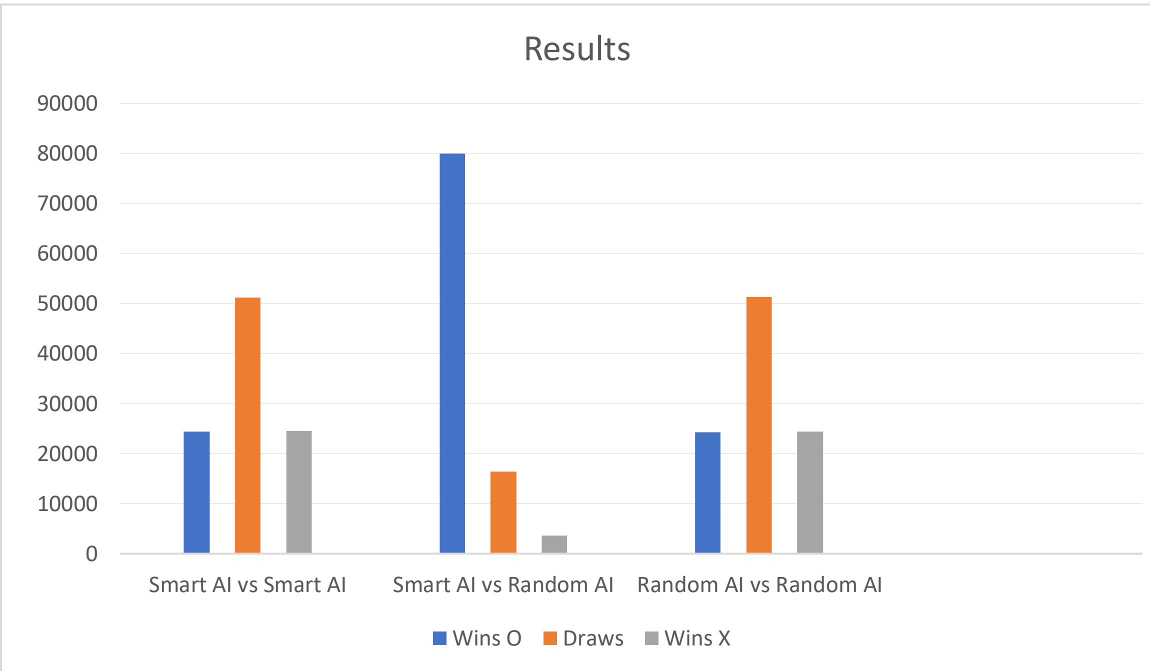
Wins Smart AI 1 (O)	=	24349
Draws	=	51172
Wins Smart AI 2 (X)	=	24479

2) Smart AI vs Random AI:

Wins Smart AI (O)	=	79943
Draws	=	16464
Wins Random AI (X)	=	3593

3) Random AI vs Random AI:

Wins Random AI 1 (O)	=	24314
Draws	=	51337
Wins Random AI 2 (X)	=	24349



Conclusion

After looking at the data, it is clear that the Smart AI is a lot better than the Random AI, but the number of draws did initially surprise me, however, it all made sense after considering what I wrote in the hypothesis, if the Random AI got to play first, it is a bit more likely that the game will end in a draw or even the Random AI winning.

The same applies to Smart AI vs Smart AI, the differences in the number of wins are purely because one of them has gotten more turns to start the game.

While Random AI vs Random AI follows a very similar trend, we cannot conclude that the differences in the number of wins are entirely because of one of them getting more turns to start the game, this is because the placement of ‘X’ or ‘O’ is random and is not tied with an algorithm.