Report 4

- ***Name:*** *Shikhaliyev Anar*

Project Title:

Design and development of a board game that incorporates elements of tactical decision-making and strategic planning
What are you going to do?
I am going to design and develop a board game that enables players to play against one another and most importantly play against AI. I want to analyze different AI approaches such as heuristics, tree- traversal algorithms - minimax, alpha-beta pruning, adversarial search algorithms, and other techniques to improve performance such as memorization.

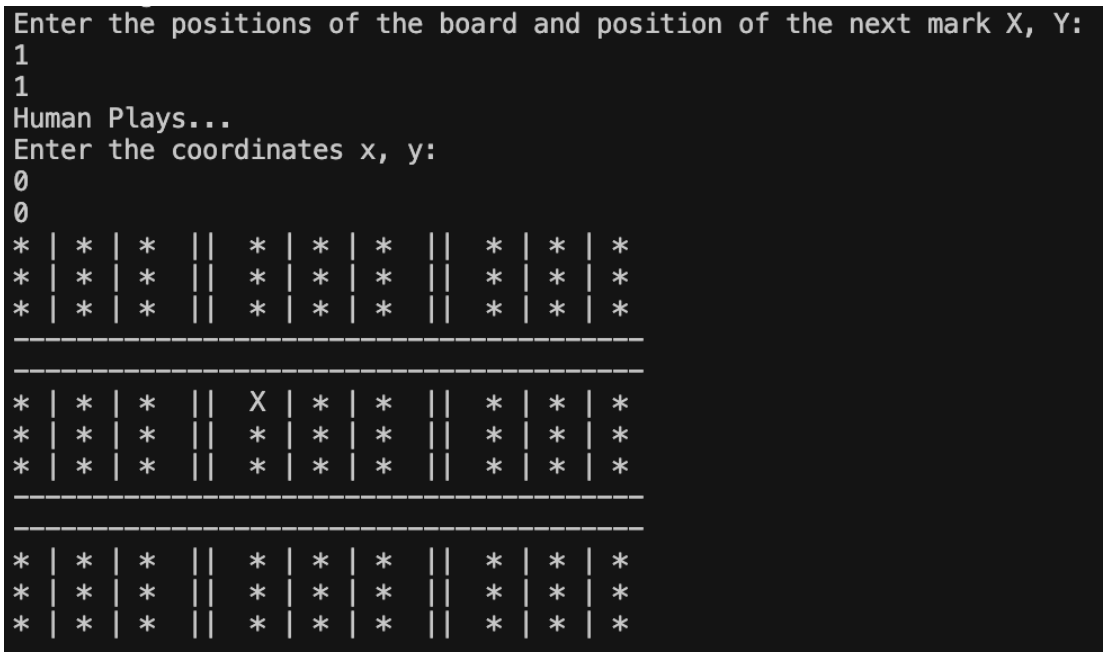
Report:

Compared to previous reports, I want to deep dive more in the technical details of how algorithm works and decides accordingly in which situations. But we first to define basic game rules.

This game is a modified version of the classic game of Tic-Tac-Toe that presents a greater level of difficulty for computational systems compared to the standard variant.
The composite board is comprised of nine smaller boards, each measuring 3 by 3 units, forming a larger board also measuring 3 by 3 units.

In this game, participants engage in a turn-based sequence, with each player taking alternating turns. The objective is to achieve victory on a smaller game board, similar to the traditional game of Tic-Tac-Toe, by strategically arranging three of one's own symbols consecutively in a row. Once a player emerges victorious on a tiny board, they proceed to place their designated symbol in the corresponding location on the larger board. The game concludes when a player successfully aligns three symbols consecutively on the huge board, or in the event of a draw when all squares have been filled.

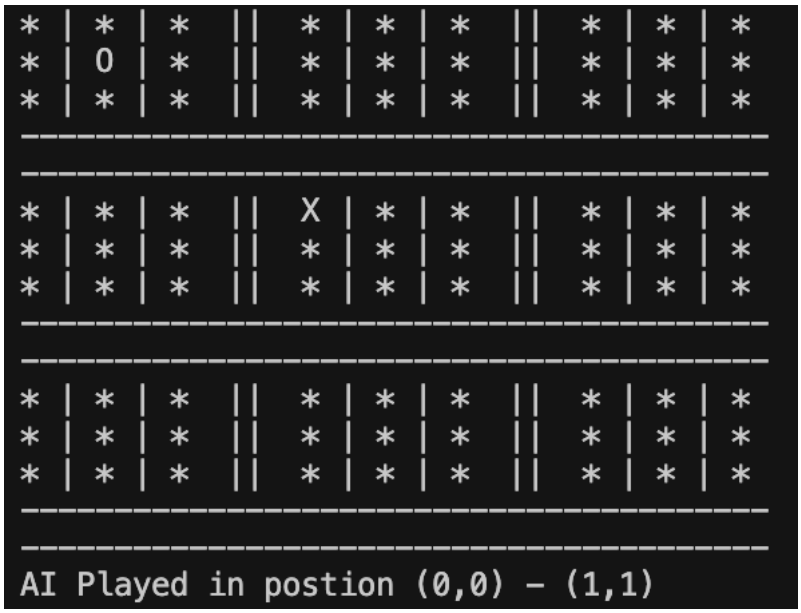
Image 1



In order to enhance the level of engagement during gaming, it is essential for a player to strategically position their symbol inside the tiny board that aligns with the location of the previous move within said small board.

As seen in Figure 1. The first player selects the central position (1,1) on the game board and thereafter decides to make a move at the coordinate (0,0) inside that specific subgrid. Subsequently, following the display of Image 2, the artificial intelligence (AI) will proceed to execute its move inside the designated tiny board located at coordinates (0,0).

Image 2



Based on the findings of the research, many AI algorithms have been included into the game in order to compete with human players. These algorithms have been tweaked to work with the board game's unique rules and features. In-game artificial intelligence algorithms included the ones such as Random agent algorithm, Minimax algorithm, alpha-beta and several heuristic algorithms. There are different variations of each of these algorithms as well. First, I've tried to build a reliable agent to play simple 3x3 Tic Tac Toe and used components of this agent such as board evaluations, heuristic in a Big Tic Tac Toe.

The Minimax method is used for traversing the whole tree, while the Minimax algorithm with alpha-beta pruning is employed to reduce the number of pathways examined by the algorithm

The evaluation of this agent is based on computing the score of a certain array (whether it be a row, column, or diagonal) by considering the count of the player's symbols (X) and the opponent's symbols (O) inside that array. The function calculates the summation of the items inside the array and afterwards determines the absolute value of this total. Subsequently, the algorithm ascertains if the array exclusively comprises symbols belonging to the player, symbols belonging to the opponent, or a combination of both. In the scenario when the array exclusively consists of the symbols representing the player, a score is assigned. This score is calculated by dividing the total of the array components by the absolute value of array, and then multiplying the result by 10 raised to the power of absolute sum of array -1. The score exhibits exponential growth as the quantity of symbols belonging to the player rises inside the array. If the array consists of a combination of symbols belonging to both the player and the opponent, or solely consists of symbols belonging to the opponent, the score is assigned a value of 0.

For the big game, there are a few heuristics that are implemented separately or as a combinations.

One of the heuristics place emphasis on the identification of a potential winning sequence for either the player or the opponent throughout the whole of the Tic Tac Toe board. The function verifies if a player has achieved victory in the game by acquiring a consecutive series of their respective symbols in either a row, column, or diagonal. In the event that the player emerges victorious, the system will output a significantly positive score of 10000, hence signifying a favourable condition for the player. In the event that the opponent emerges victorious, a very negative score of -10000 is returned, indicating an unfavourable condition for the player. In the absence of a declared winner, the system will provide a neutral score of zero.

The other heuristic places emphasis on the assessment of the smaller boards included inside the Tic Tac Toe board ***on the domain of the small board***, with the objective of allocating points according to certain patterns.

Moreover, there are also another algorithm used only in certain stages of the game which takes into account several criteria, such as the condition of smaller boards, the condition of the larger board, and the player's command over the subsequent move on a smaller board. The algorithm computes scores for each of the aforementioned criteria and yields their summation as the ultimate assessment score.