UNBEATABLE TIC-TAC-TOE

Jerome Santos

Gaudwin Timoteo

Loyola University Chicago

Department of Computer Science

COMP 460

Spring Semester, 2021

Professor: Dr. Hayward

Due: May 3rd, 2022

Table of Contents

List of Project Participants

As a group we all completed all presentation material, and the final report collectively.

Jerome Santos:

- Research on Tic-Tac-Toe and Minimax Algorithm
- Implementation of GUI through HTML/CSS/JS
    - Home Page, Human Vs. AI, AI Vs. AI, Human Vs. Human
- Implementation of Minimax Algorithm
    - Human Vs. AI, AI Vs. AI
- Application Navigation

Gaudwin Timoteo:

- Research on Tic-Tac-Toe and Minimax Algorithm
- GUI design
- Implementation of GUI through HTML/CSS/JS
    - Human Vs. AI
- Implementation of Minimax Algorithm
    - Human Vs. AI
- Application Navigation

Abstract of the Project

There are many games out there that use many different algorithms to create a stable "Artificial Intelligence". Some games that are commonly made for simple applications are sometimes too easy. One of those games is typically Tic-Tac-Toe. Tic-Tac-Toe is a widely-known strategy game, but what one does not know is that the game actually is a game of how not to lose. In the game of Tic-Tac-Toe, "the second player always reacts to the first." The second player only has the option to defend, otherwise they will end up losing. If the first player makes no errors, the second player will have no choice on where they move. This game consists of playing for the draw or the loss. Most of the time, the second player still plays the game knowing they will not win, unless the first player makes a wrong move. Our application, Unbeatable Tic-Tac-Toe, goes hand-in-hand with that idea where the second player will try to draw the game or win the game if player one makes a mistake.

Project Narrative

Our main goal is to provide a web application game that utilizes the minimax algorithm. To start with, we created the Tic-Tac-Toe algorithm that has three game options: AI vs AI, Human vs AI, and Human vs Human. We aimed to make this two player game to be playable by one person by providing an opponent using minimax. We named it unbeatable to make a point that the only possible outcome while playing with the AI is either draw or the human player loses.

In our future updates, we plan on adding more games that make use of the minimax algorithm. On top of our list is Mancala, a two player strategy-based game with the objective of collecting pieces from the opponent. We are also targeting to come up with an option of simulation mode. This option would help the user to strategize better by replaying the every-case scenario that could possibly happen in the real game.

Design Specifications and Considerations

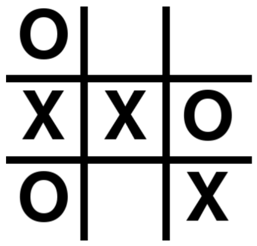
Our group had gone through various ideas for this project. At the beginning of our project planning, we had decided to choose an algorithm first, and figure out how to implement that algorithm. A few of the choices we had looked into were the Tower of Hanoi with a recursion algorithm and then Tic-Tac-Toe or Mancala with a minimax algorithm. After careful consideration, we both have decided to choose one of the games that utilize the minimax algorithm. By recreating Tic-Tac-Toe and then implementing the minimax algorithm, we would be able to clearly learn and explain how the algorithm works within our project.

We started by learning how Tic-Tac-Toe works and making sure we understood the main rules for the game. The game rules go as follows: “

- In a 3-by-3table, the “O” player always goes first.
- Players will take turns placing the Xs and Os on the table until either player has three in a row (vertically, horizontally, or diagonally) or until all spots on the table are completed.
- If a player is able to mark down three Xs or three Os in a row, then that player wins.
- If all spots on the table are filled and neither player has completed a full row of Xs or Os, then the game is a tie.”

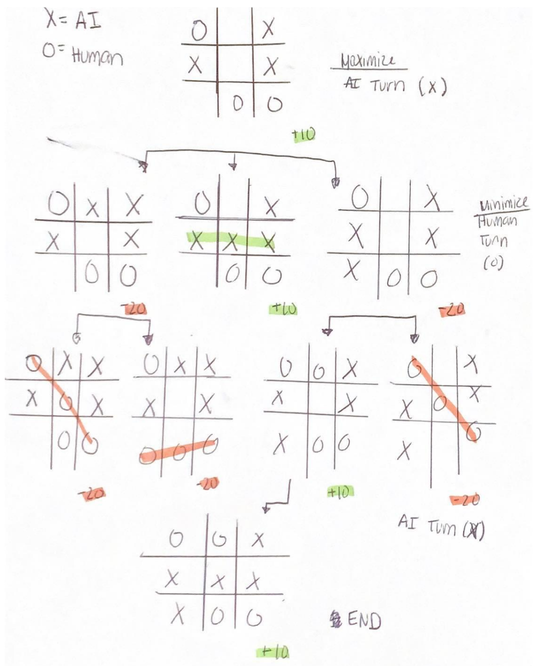
(Erik Arneson)

With the rules in mind, we were able to code an average Tic-Tac-Toe game. This Tic-Tac-Toe game had almost no real functionality and only was able to press a single O or X character onto the table. This also had a very simple graphic user interface as seen in Figure 1.



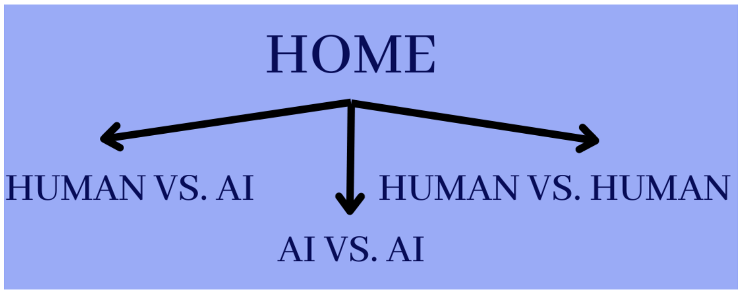
(Figure 1)

After working on the basics for Tic-Tac-Toe, we went ahead and analyzed how we were going to implement the minimax algorithm. Minimax can be defined as a kind of backtracking algorithm that is implemented a lot in game theory and decision making to find the optimal move for a player. We went through many documents and videos to see how minimax works with Tic-Tac-Toe just so we could get a better understanding of how to implement it to our own project. During the process of trying to understand minimax, we drew a few scenarios to outline how minimax works just so that we could visually see the outcome we wanted, as seen in Figure 2.



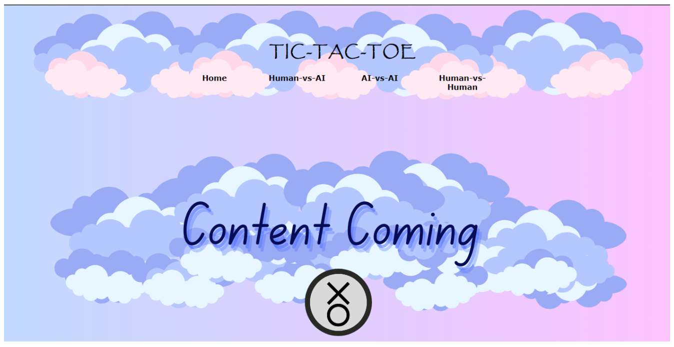
(Figure 2)

After working on the Minimax algorithm and fully understanding how to incorporate it, we worked on updating the overall design of the application and worked on the navigation of the application. The navigation starts off at the home screen. There are then three main screens in this iteration of the application. There is the option for Human versus AI where the human player will play against the Minimax algorithm. There is the option to watch the AI versus AI where its just the Minimax Algorithm against itself. Finally there is the Human versus Human option. As seen in Figure 3, it is a very simple navigation as there is no real need to go beyond as we are trying to showcase just the Tic-Tac-Toe game.

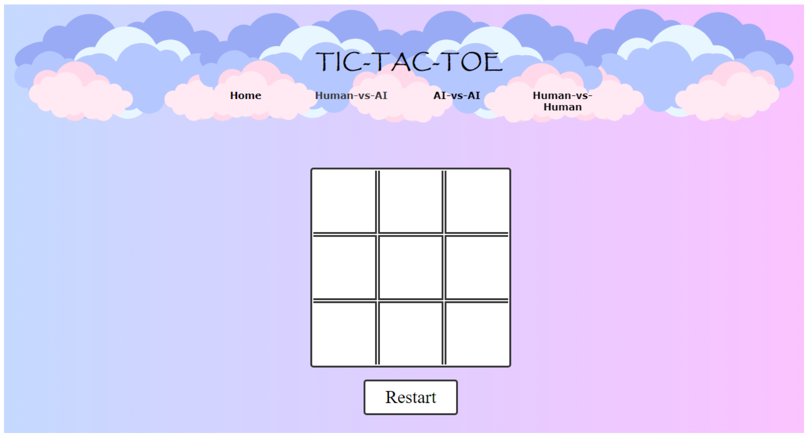


(Figure 3)

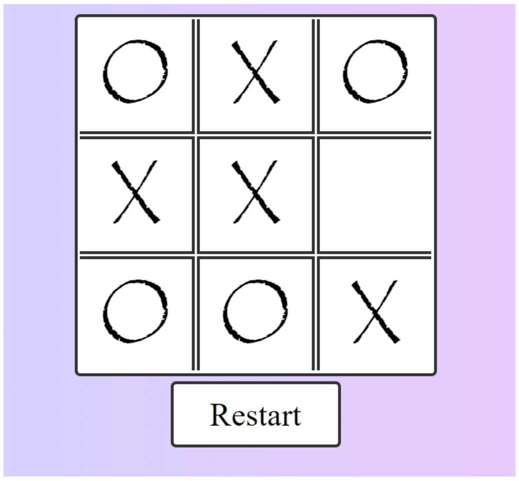
We took the creative liberty to make our designs colorful and bright. The sunset/sunrise scheme with the blue hues and subtle pinks and purples compliment each other to give off a relaxing essence to a game that is unbeatable and eventually be frustrating.



Each of the tabs mentioned in the navigation leads to the Tic-Tac-Toe table (as seen in Figure 5) which leads to it being consistent among the tables per mode of game. The design throughout stays consistent with the blues, purples, and pinks and then the fonts we used to make it look more like chalk as Tic-Tac-Toe is usually a kids game and a lot of kids use chalk on the side-walk to play Tic-Tac-Toe like in Figure 6.



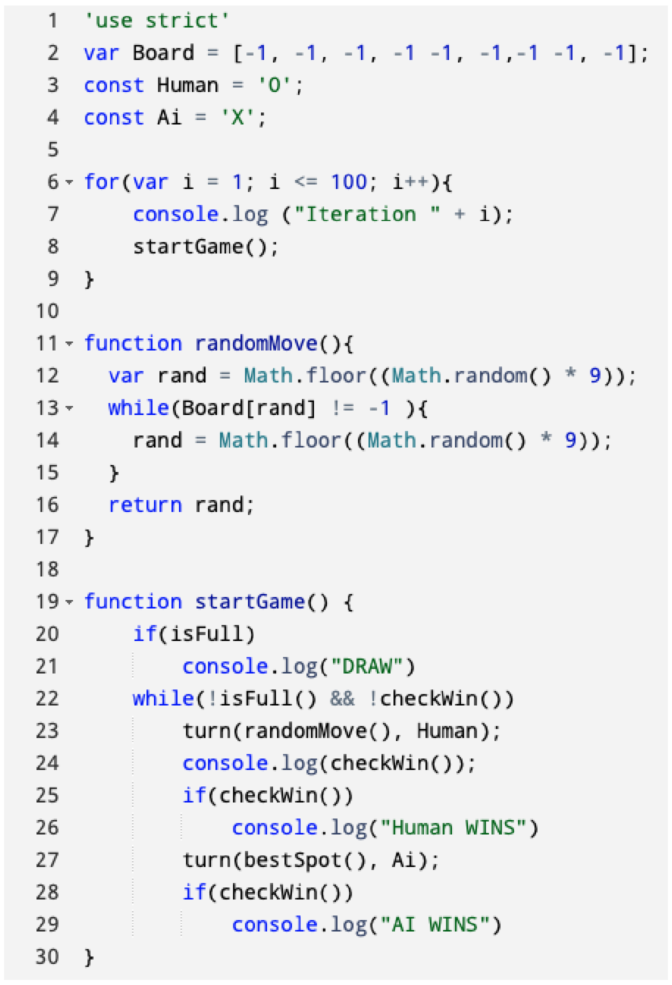
(Figure 5)



(Figure 6)

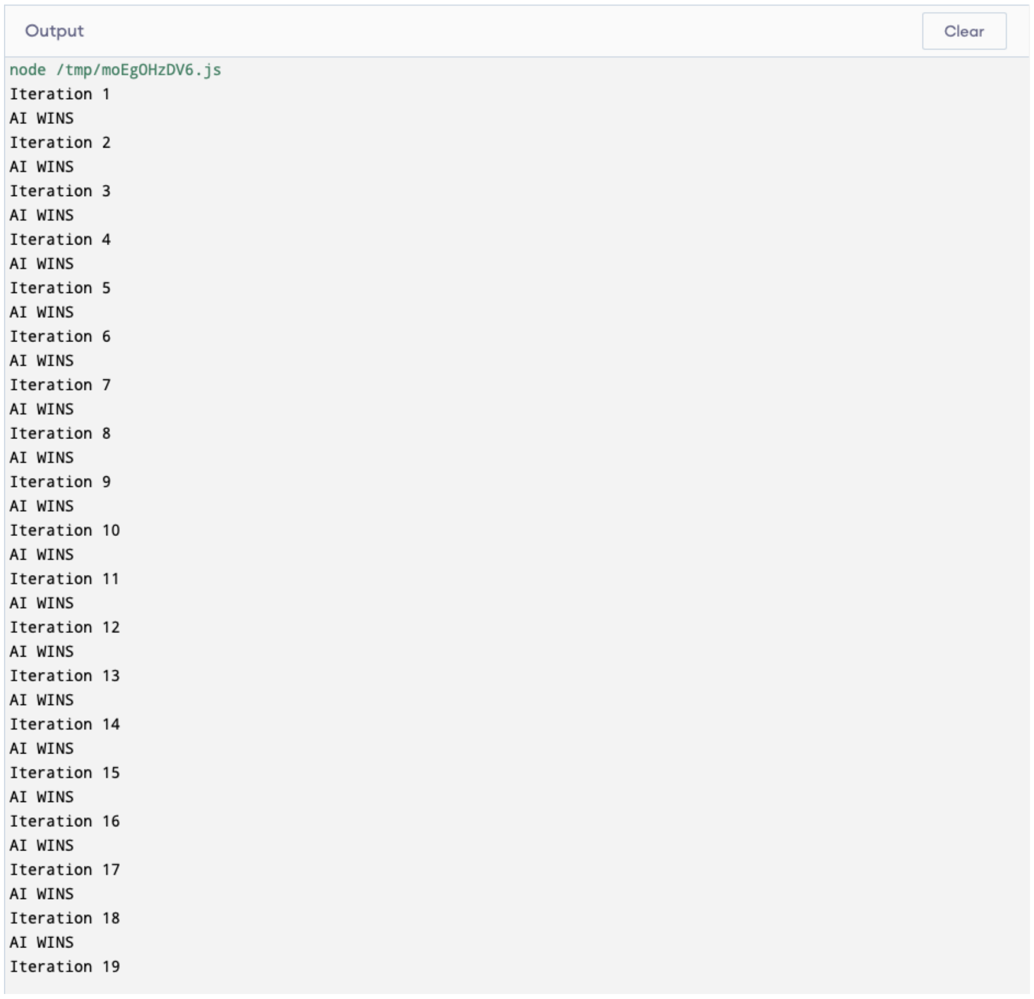
Testing and iterative dsign

Throughout the development, we have played the game as we add new features. Gradually testing the functions helped us to ensure that each function is behaving properly as we intended to. As we add new functions, we are conducting the testing regularly from the beginning to guarantee that the recent update did not run into the current version. Due to the lack of participants, we were not able to conduct many iterative tests with different people. To compensate with that, we created a program that generates a random move to play against the AI.



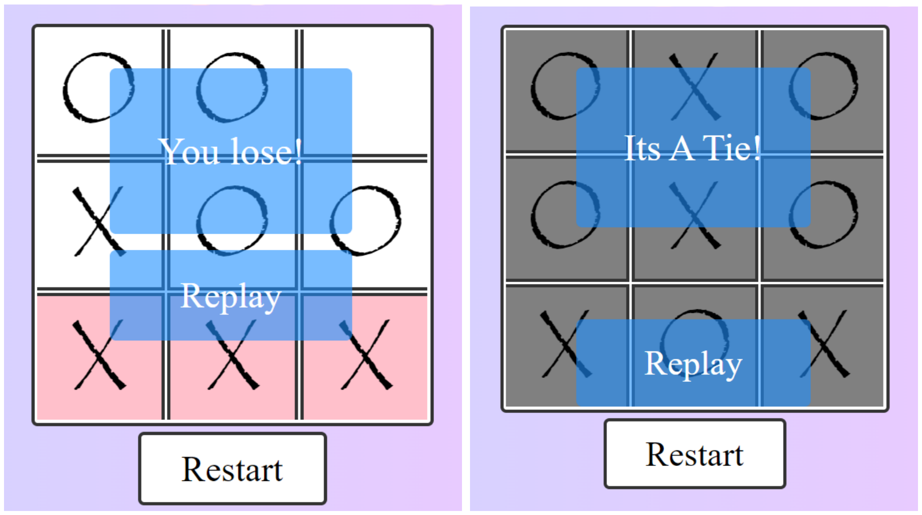
```
'use strict'
var Board = [-1, -1, -1, -1 -1, -1,-1 -1, -1];
const Human = 'O';
const Ai = 'X';

for(var i = 1; i <= 100; i++){
    console.log ("Iteration " + i);
    startGame();
}

function randomMove(){
  var rand = Math.floor((Math.random() * 9));
  while(Board[rand] != -1 ){
    rand = Math.floor((Math.random() * 9));
  }
  return rand;
}

function startGame() {
    if(isFull)
        console.log("DRAW")
    while(!isFull() && !checkWin())
        turn(randomMove(), Human);
        console.log(checkWin());
        if(checkWin())
            console.log("Human WINS")
        turn(bestSpot(), Ai);
        if(checkWin())
            console.log("AI WINS")
}
```

To certify the main goal of the application that the minimax algorithm is unbeatable, we created a 100 iteration. The code that we made will generate a random number to represent an index number in the board. The program will keep generating a number until the game is over. The result shown solidifies our claim as the result showed that the AI won in every iteration.



```
node /tmp/moEgOHzDV6.js
Iteration 1
AI WINS
Iteration 2
AI WINS
Iteration 3
AI WINS
Iteration 4
AI WINS
Iteration 5
AI WINS
Iteration 6
AI WINS
Iteration 7
AI WINS
Iteration 8
AI WINS
Iteration 9
AI WINS
Iteration 10
AI WINS
Iteration 11
AI WINS
Iteration 12
AI WINS
Iteration 13
AI WINS
Iteration 14
AI WINS
Iteration 15
AI WINS
Iteration 16
AI WINS
Iteration 17
AI WINS
Iteration 18
AI WINS
Iteration 19
```

Something that has improved along the way is definitely the design of the table for the Tic-Tac-Toe game. We went from a very basic Tic-Tac-Toe board to a fully reactive table where it highlights the winner or highlights a tie. This is to help the players easily distinguish the outcome of the game and overall helps with the game's aesthetic so that it is not just a static table that does nothing.



(Figure 7)

Restrictions, limitations, and constraints

The usage of the minimax algorithm only provides us a definite amount of moves. This causes the moves of the AI vs AI to be the same for every iteration. Due to the lack of time, we were not able to develop with the better randomization of the first move. In our future updates, we plan on having a function that randomizes the first move to any of the corners in the board. Additionally, we would have liked to include a variety of games such as mancala and tower of hanoi and checkers. However, with the lack of time, we dedicated ourselves to focusing on one game and fully understanding how the Minimax algorithm is to work with it.

Conclusion

Despite there being a lot of aspects and elements that go into creating an algorithmic based application, ranging from the actual build of the site to the user interface, the goal of any application should not be lost in the work of making it. With Tic-Tac-Toe, our application's overall goal is to demonstrate our understanding and implementation of the Minimax algorithm into a working application. While working on the planning and development phase of our application, there were many layers and elements that we had to discuss and think through. These aspects included the design of the application, the flow of the pages, which packages and algorithms to implement and other design decisions. While there were many style choices to pick from, the purpose of our project could have been easily lost in the process of creating the application.. Despite having some roadblocks along the way, we believe that we have made an application that showcases our original ideas. The various elements that could have misdirected the application actually assisted us in developing and improving it to be more of what we had envisioned. Thanks to the feedback that we have received over the system from our classmates and professor, we were able to develop an application that hopefully meets the needs of its users.

Works Cited

Arneson, Erik. “Learn Tic-Tac-Toe Game Rules with Variants.” *The Spruce Crafts*, The Spruce Crafts, 12 May 2020,
https://www.thesprucecrafts.com/tic-tac-toe-game-rules-412170.