SENG 465

Artificial Intelligence in Game Programming

Project Report

Eyüp Öztürk

201928026

Tic-Tac-Toe Game with Minimax Algorithm

Instructor

Prof.Dr. Mehmet Reşit Tolun

1. Summary

I prepared my project using Tic Tac Toe game. I made the game in my project using the Unity 3D game engine. I used a game engine because I knew that I would get the best results both visually and mechanically by using a game engine. I used elements such as UI, sound, animation, code very easily. In the game, we have the chance to play both with artificial intelligence and normally. I used the minimax algorithm for artificial intelligence, artificial intelligence kicks in against our every move and makes the best possible move, as a result, we either draw or the artificial intelligence wins.

2. Previous Work in The Literature

Since I made the game with the Unity game engine, I chose the games I would compare from the games made with unity.

2.1 First Game

Although the first game is not very good in terms of UI, it is very well designed mechanically, so it made the coding part complicated and long, the code I wrote is shorter and more understandable. It did not collect the minimax algorithm in a single function but distributed it to other functions. I can do it in a single function and call it any way I want.

url: https://github.com/BadToxic/ttt

2.2 Second Game

The second game is made in 3d instead of 2d compared to other games, so its gameplay and appearance are very different. Not much attention has been paid to the UI part, but the code part has been designed very well. Artificial intelligence did not make the algorithm in a single function in the same way.

URL:https://github.com/codehoose/tic-tac-toe-live-stream

2.3 Third Game

The last game was made with Python instead of Unity. That's why the UI part looks worse than the game I made. It uses minimax algorithm as artificial intelligence; it is combined in one function like I did. The codes are neat and well written, so it is better readable. It performs better mechanically because it is written in Python

URL: https://github.com/AlejoG10/python-tictactoe-ai-yt

3. Description of the Project

My project consists of two stages, the first stage is to integrate the game into the unit and to use artificial intelligence while keeping the main theme of the game, and the second part is to integrate artificial intelligence into the game using one of the algorithms we saw in the lesson. It took a lot of time to build and combine them, but using the artificial intelligence algorithm took less time than I thought. I tried to do my best. I faced a lot of difficulties both for the UI part and for the coding to work properly.

3.1 implementation

I used c# coding language because I made the project using unity. First, I made the valid coding for unity. The game consists of two different scenes, I combined them together. The first scene is the starting scene of the game and there are two buttons from this scene. With these buttons, we can make two different choices, either with artificial intelligence or by ourselves. This first scene will be unlocked each time the game is abandoned. I made the main mechanic of the game from the second scene, I used many features of unity here. I put in a lot less effort than I would normally because it's very easy to do this kind of thing with unity.

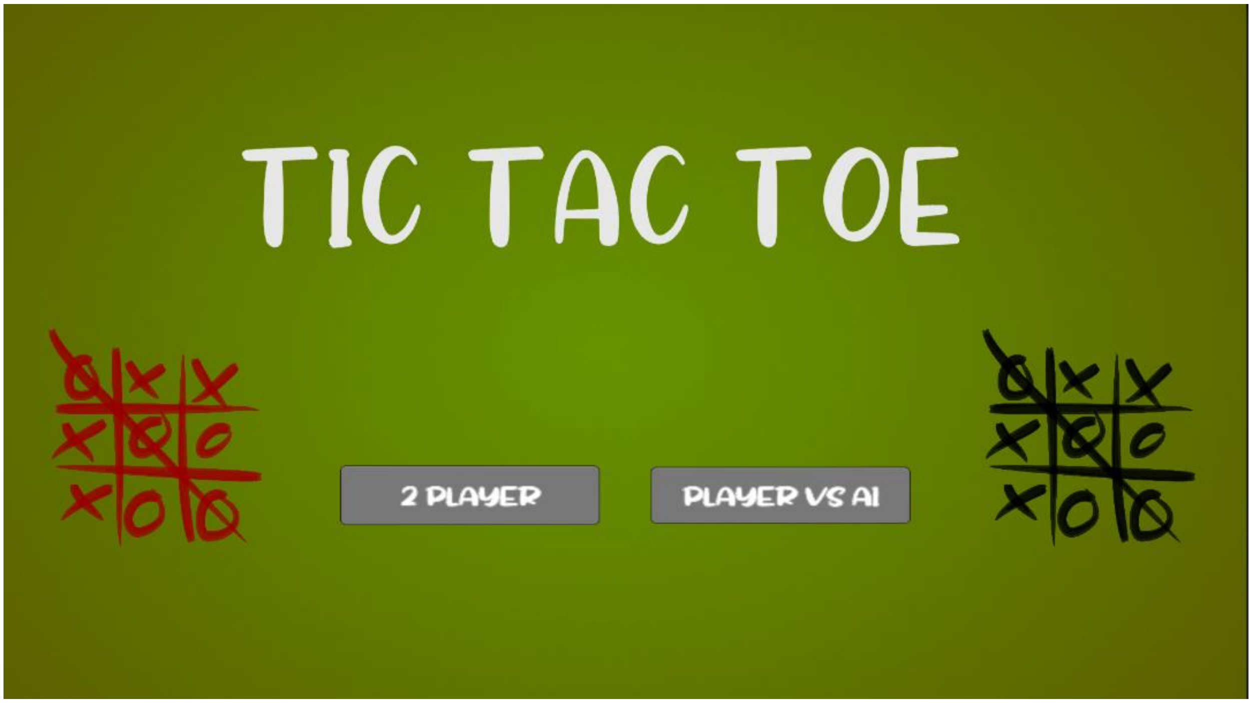
Since the Tic Tac Toe game consists of 9 different sections, I put a button in each section, I received the value of each pressed button, and in this way, I was able to convey my used move to the artificial intelligence. Artificial intelligence, on the other hand, makes its own move instantly using the Minimax algorithm.

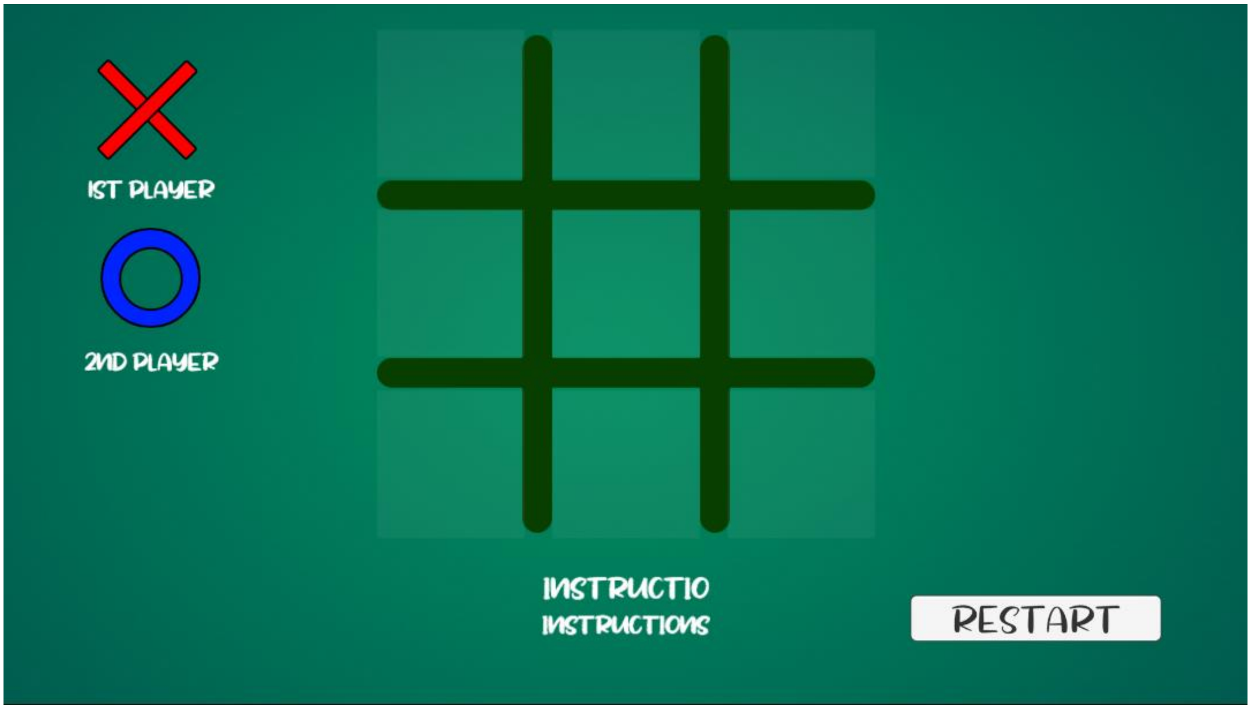
3.2 Difficulties Encountered

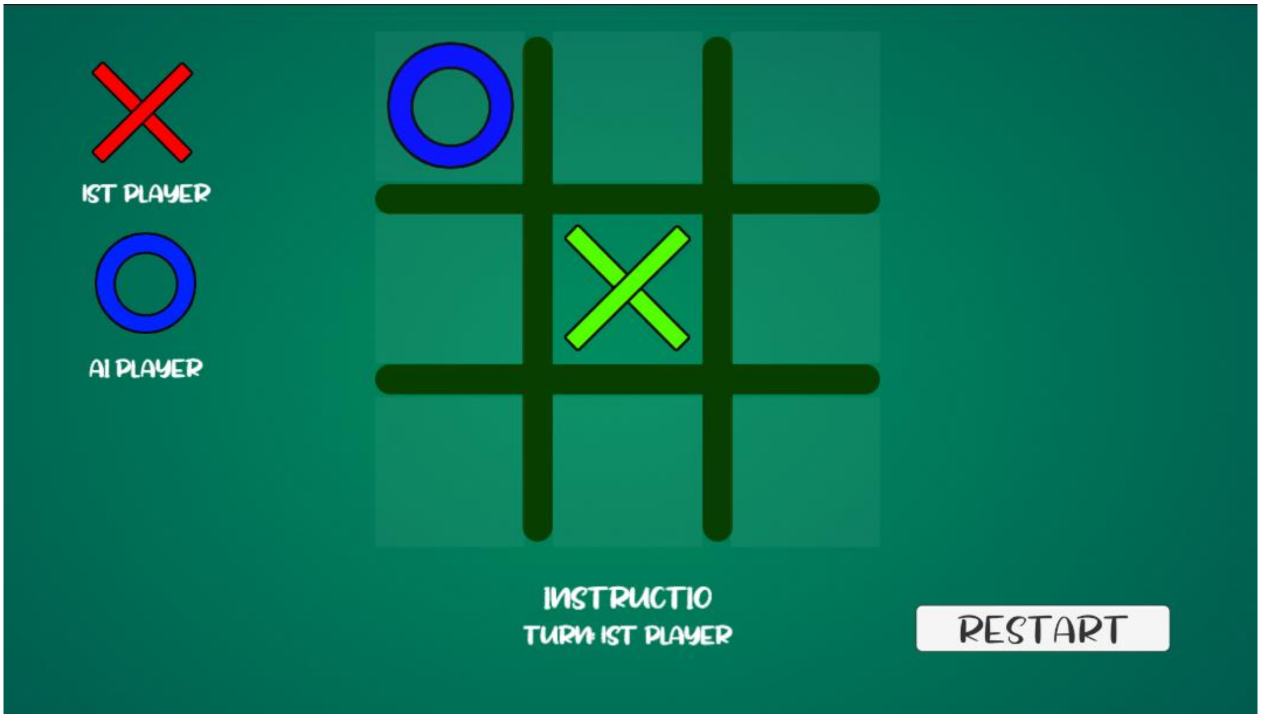
The most difficult part I encountered while making the project is to organize the project according to unity and bring it into a working state. Using the Minimax algorithm for the Tic-Tac-Toe game is one of the easiest ways to use it, so I did not encounter much difficulty while designing the algorithm or integrating it into unity. I used pictures and different sounds for in-game visualizations, I just had a bit of a hard time finding them.

4. User Interface

4.1 Run-Time Screenshots

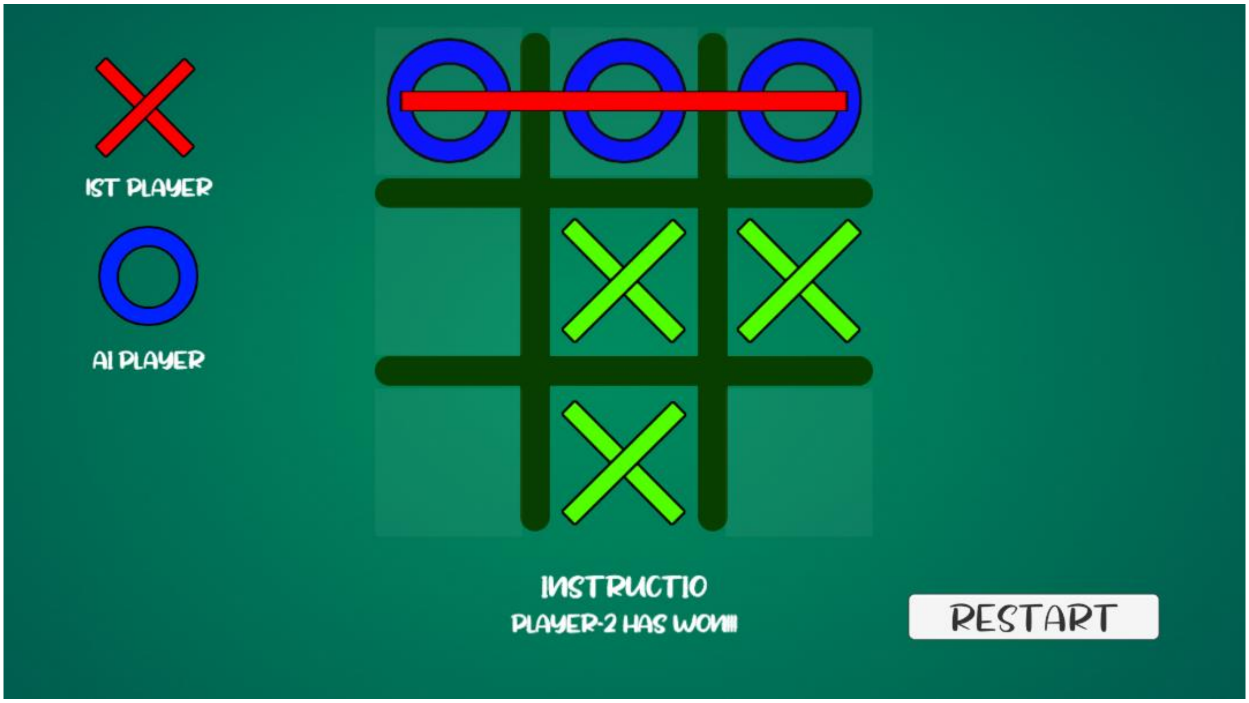

TIC TAC TOE
2 PLAYER
PLAYER VS AI


1ST PLAYER
2ND PLAYER
INSTRUCTIO
INSTRUCTIONS
RESTART



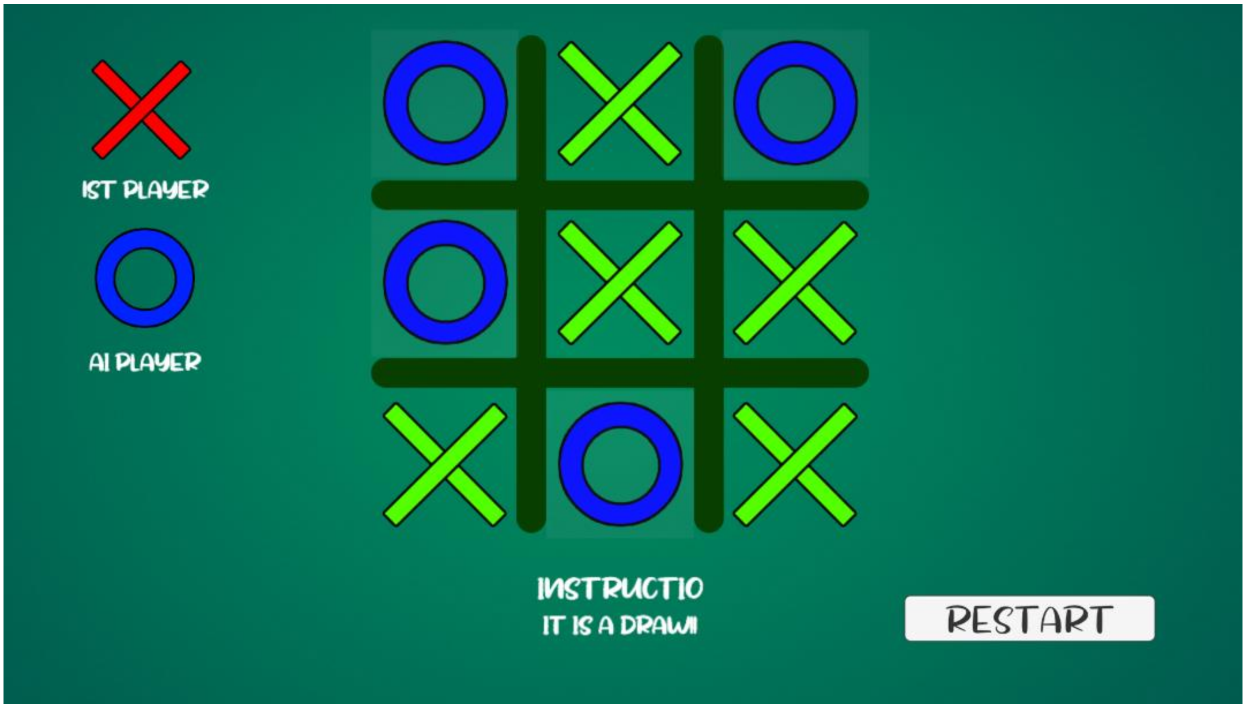
INSTRUCTIO

TURN: 1ST PLAYER

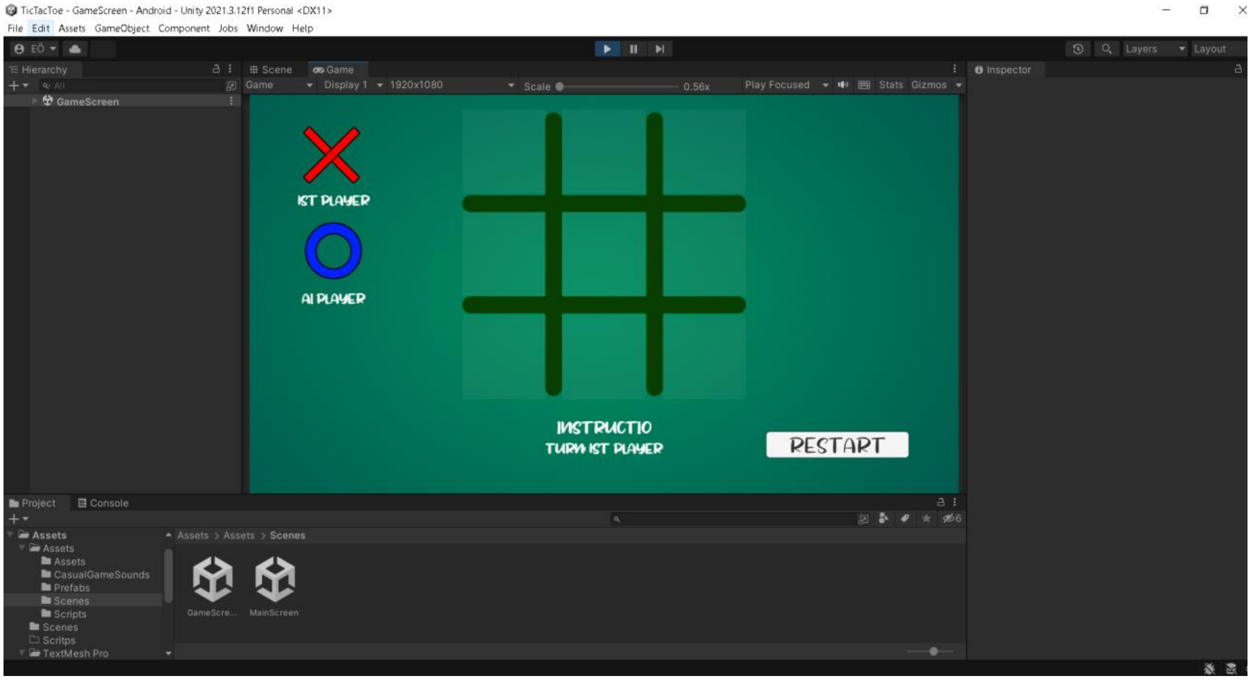


INSTRUCTIO

PLAYER-2 HAS WON!!!



INSTRUCTIO
IT IS A DRAW!!



INSTRUCTIO
TURN IST PLAYER

4.2 Final Comments on the Project

I believe that I have done both the construction phase, coding, and design of the project properly, I think that I have used the artificial intelligence algorithm properly and integrated it into the unity game engine well. Since I have been playing Tic Tac Toe game since I was little, I chose to use this game in my project, so it was a nice work for me.

References:

- Unity Documentation
- Geeksforgeeks web site
- JavaTpoint web site
- WikiHow web site

Appendix:

My project: https://github.com/softflied/TicTacToe.git