

3D TIC TAC TOE WITH

64 SLOTS

ITCS 6150 Intelligent Systems

-BY DR. ZBIGNIEW W RAS

**Submitted By (Team 9):**

**Gaurav Avula – 801319867**

**Leela Santoshi Totapally – 801273358**

**Mahija Shivani Dadisetty - 801303384**

**Sarath Sri Sai Chennu - 801307322**

**INDEX**

* Project Description… 3
* System Flow Diagram……………….……………………....................………………………………………………….4
* Data Flow Diagram……………….………………………....................……………………………………….………….4
* Input Design ……….……………….………………………....................……………………………………………….….4
* Output Design ……….…………….………………………....................………………….……………………………….4

**Project Description:** The main purpose of this project is to implement the 3-dimensional TIC-TAC-TOE with 64 slots using minmax method with alpha-beta pruning with three difficulty levels easy, difficult, and insane with their respective looks ahead. Human and a computer will compete in the game.

The game begins with one of the players choosing their character. To create their respective cells, one player selects ‘O’, while the other selects ‘X’. The player will also get to select the input difficulty and gets to select who plays the game first. The difficulty levels are:

* Easy
* Difficult
* Insane

One of the players tries to block the square either in a row, column, or diagonally, if the opponent has three in a row. The game ends when one of the other players has filled a row, column, or diagonal through board or through boards with either the character ‘X’ or ‘O’. The player can choose to play the new game at any moment while the game is being played.

When either the computer or the person has four in a row, the player wins the game and the scores for both are updated and displayed. Due to the special properties of 3D Tic-Tac-Toe, a tie cannot occur, hence a tie counter is not necessary.

In this project, we will use minmax method with alpha-beta pruning to implement the decision tree. This is a search algorithm that seeks to reduces the evaluation of the nodes. The degree of difficulty directly determines how many times the machine looks ahead perform up to 6. If the player selects difficult level as ‘Easy’ then 2 looks ahead are performed, if the ‘Medium’ difficulty level is selected then 4 looks ahead are performed and if ‘Hard’ difficulty level is selected then 6 looks ahead are performed.

The user may pick who goes first (Human by default), whether they want to be "X" or "O" (X by default), and what level of difficulty they want to play on (Medium by default). By adjusting how far into the move tree the algorithm looks ahead, different levels of difficulty can be achieved. This is integrated with a user-friendly Graphical User Interface.

**System Flow Diagram:**

**Diagram

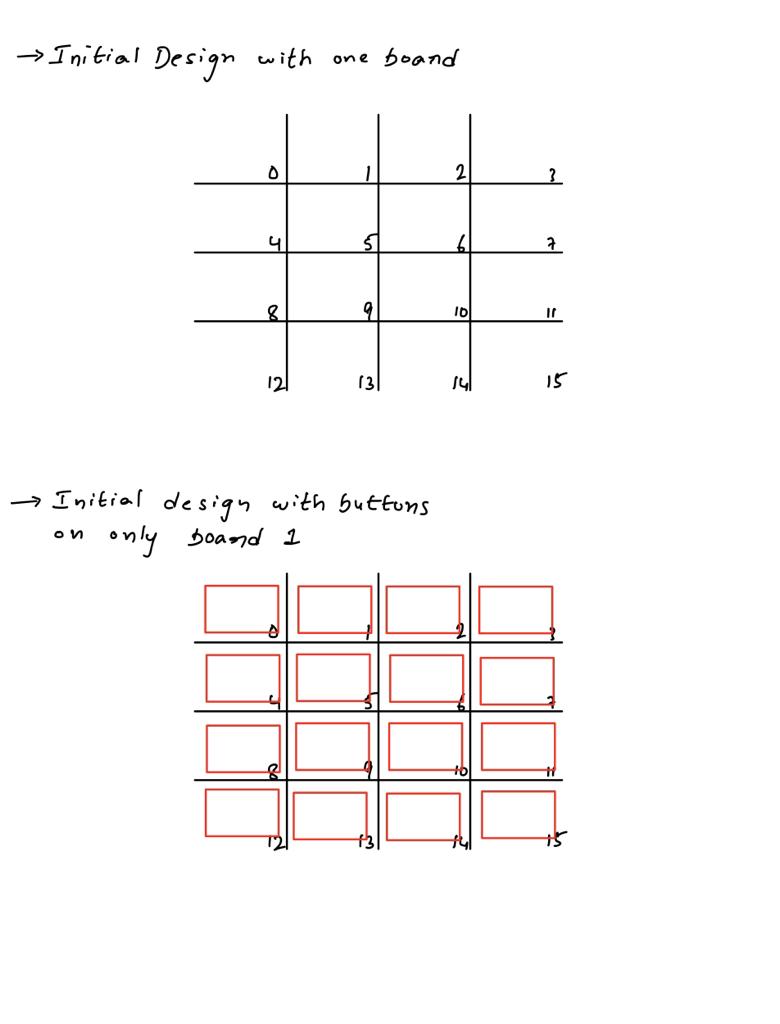
Description automatically generated**

**Data Flow Diagram:**

**Diagram

Description automatically generated**

**Input Designs:**

****

**Diagram, table

Description automatically generated**

**A diagram of a house

Description automatically generated with low confidence**

**Output Design**

A picture containing chart

Description automatically generated

***Output design for Tic Tac Toe with 64 slots:***

A picture containing antenna

Description automatically generated

***TIC-TAC-TOE Graphical User Interface***

***A screenshot of a graph

Description automatically generated with low confidence***

***TIC TAC TOE GAME:***

***Text

Description automatically generated with low confidence***

***AI WON:***

***Line chart

Description automatically generated with low confidence***

***HUMAN WON:***

***A picture containing diagram

Description automatically generated***