

# Artificial and Computational Intelligence

## Assignment 2

### Problem statement : Gaming

#### **Title: Implement a Two-Player Connect Four Game with Fixed Depth Minimax**

You are required to develop a Simple-Connect-Four game on a 7x6 grid where a human plays against a computer.

#### **Requirements:**

**Grid Representation:** Represent the 7x6 grid using a 2D array.

**Human Player:** Allow the human player to select a column to drop their disc. Ensure that the selected column is not full.

**Computer Player:** Implement the Minimax algorithm for the computer's moves with a fixed depth of 3. The algorithm should consider factors like winning, blocking, and strategic positioning.

**Game Logic:** Alternate turns between the human and the computer. After each move, update the grid and check for a win (four in a row horizontally, vertically, or diagonally) or a draw (all columns are filled).

**Output:** Display the grid after each move and announce the winner or declare a draw when the game ends.

#### **Rules to Play Connect Four:**

1. **Objective:** Connect four of your discs in a row horizontally, vertically, or diagonally on a 7x6 grid.
2. **Turns:** Players take turns to drop discs into one of the seven columns. The disc falls to the lowest available slot in that column.
3. **End Game:** The game ends when a player connects four discs in a row or the grid is full, resulting in a draw.

#### **Sample Moves:**

- **Move 1:** Player 1 drops a disc into column 3; it falls to the bottom.
- **Move 2:** Player 2 drops a disc into column 4; it falls to the bottom.
- **Move 3:** Player 1 drops another disc into column 3; it stacks on top of their first disc.

a. You are free to choose your own static evaluation function. Justify your choice of static evaluation value design and explain with a sample game state. Do not use any machine learning model for the evaluation function.

b. In the Connect Four game, one player must be a human and provide dynamic inputs during gameplay. The other player will be simulated by the program using the Minimax algorithm with a fixed depth. The game should alternate turns between the human player and the computer, displaying the updated game state after each move.

c. Implement Python code for the design under part a, using Minimax Algorithm.

**Important Note:**

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word document for explaining the theory part. Do not include the theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be Considered equal partners and will be penalized severely.