Tic-Tac-Toe AI with Alpha–Beta and Evaluation Modes

Objective:
Build an AI that plays Tic-Tac-Toe using **Alpha–Beta pruning**, with **two evaluation options**: a classical heuristic and a ML model. The AI should support **multiple difficulty levels**, and the **user chooses** which evaluation function to use when playing.

Project Requirements

1. Game Environment

- Implement a **3×3 Tic-Tac-Toe board**.
- Implement functions for:
  - Generating legal moves
  - Detecting terminal states (win/loss/draw)

2. Alpha–Beta Search

- Implement **Alpha–Beta pruning** to select the best move.
- The search uses **an evaluation function**, which can be either:
  1. **Classical heuristic** (hand-coded)
  2. **Machine-learned evaluation**

3. Evaluation Functions (Students Must Implement Both)

- **Classical Heuristic:** Hand-coded evaluation
- **Machine-learned Evaluation**

  - Train an **AI model** using the provided dataset
  - Features include:
    - Number of X marks, O marks
    - Number of rows/columns/diagonals where X or O is close to winning
    - X in center or corners
  - Labels: +1 if X eventually wins, −1 if O wins.

4. Difficulty Levels

- The **user chooses**:
  - **Difficulty level: Easy, Normal, Hard**
  - **Evaluation function** (classical or ML)

5. User Interaction

- GUI or console interface.
- Allow the user to:
  - Choose X or O
  - Select difficulty
  - Select evaluation function
- Display the board after each move.
- Show AI evaluation scores for each possible move.