**AI Project Documentation**

**Tic-Tac-Toe**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **علي الدين اشرف حسن قطب** | **20220296** |
| **عمر احمد عطية امين** | **20220310** |
| **عمر وجيه محسن احمد** | **20220324** |
| **مريم عمرو صلاح الدين** | **20220468** |
| **ملك علاء محمد** | **20220503** |

**Introduction and overview**

**Project idea and overview**

The project aims to develop an intelligent Tic-Tac-Toe player that plays against humans in the classic 3X3 grid using advanced AI algorithms to make the optimal decisions

The main algorithms used are Minmax algorithm, alpha beta pruning and some heuristic functions

**Similar applications**

Several applications provide Tic-Tac-Toe games with AI opponents:

1.Google’s Tic-Tac-Toe Game: A browser-based game with adjustable difficulty levels.

2.Tic-Tac-Toe Android/iOS Apps: Mobile apps, often featuring user-friendly interfaces and adjustable AI strategies.

3.Command-Line Implementations: Many educational projects provide text-based Tic-Tac-Toe games to demonstrate AI concepts.

These implementations primarily rely on the Minimax algorithm or its variants for decision-making. The proposed project extends upon this with additional optimizations like symmetry reduction and heuristic evaluation.

**Literature review**

1."Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig

Describes foundational concepts of Minimax, Alpha-Beta Pruning, and heuristic design for AI decision-making.

2."Game Theory and AI Applications" by Shoham and Leyton-Brown

Explores the theoretical basis of decision-making algorithms in games.

3."Heuristics: Intelligent Search Strategies for Computer Problem Solving" by Judea Pearl

Explains heuristic evaluation functions and their role in game-playing algorithms.

4."Algorithms for Tic-Tac-Toe" by Claude Shannon

Focuses on classical approaches to solving Tic-Tac-Toe.

5.Research on Symmetry Reduction in Game Trees (Various articles from Google Scholar)

Discusses how exploiting symmetry can optimize the search space in grid-based games.

. <https://www.geeksforgeeks.org/finding-optimal-move-in-tic-tac-toe-using-minimax-algorithm-in-game-theory/>

. <https://medium.com/@amadi8/tic-tac-toe-agent-using-alpha-beta-pruning-18e8691b61d4>

. <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/>

<https://www.whitman.edu/documents/academics/mathematics/2019/felstiner-guichard.pdf>

<https://www.researchgate.net/publication/343021763_An_Intuitive_Implementation_Of_Alpha-Beta_Pruning_Using_Tic-Tac-Toe>

**Proposed solution**

**Main functionalities :**

**.** User chooses desired algorithm

**.** AI plays optimal moves using chosen algorithm

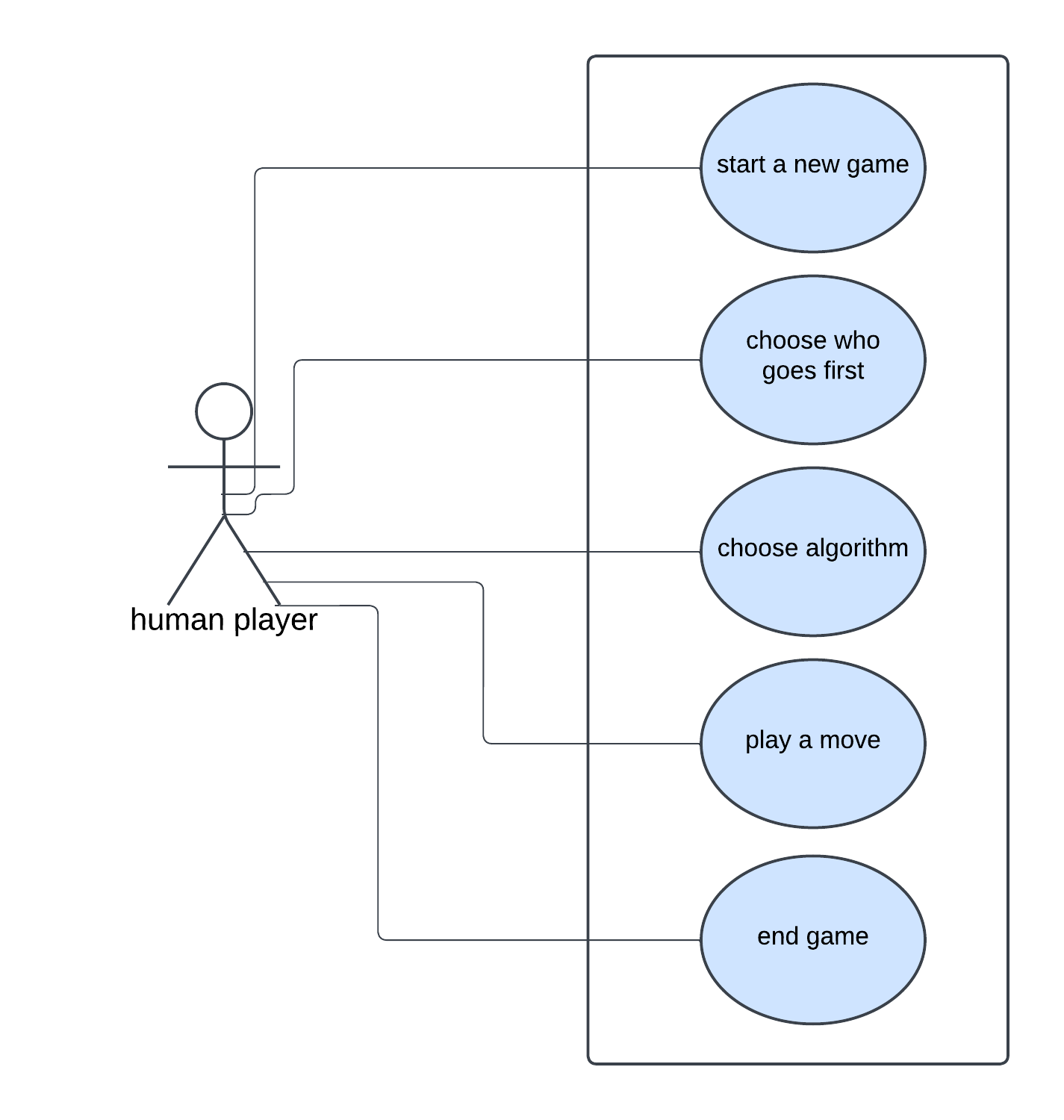
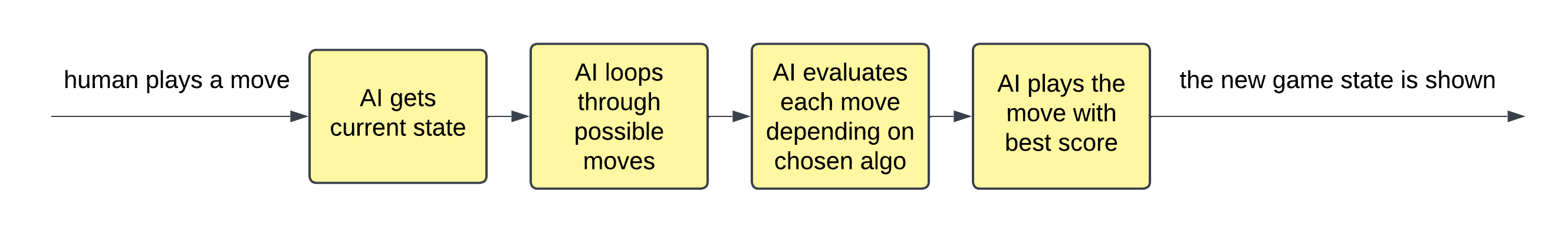
**.** Interactive GUI viewing the game board

**.** AI uses heuristics or symmetry reduction if user choose that

**.** Dark mode support

**.** Game restart option to start a new game any time

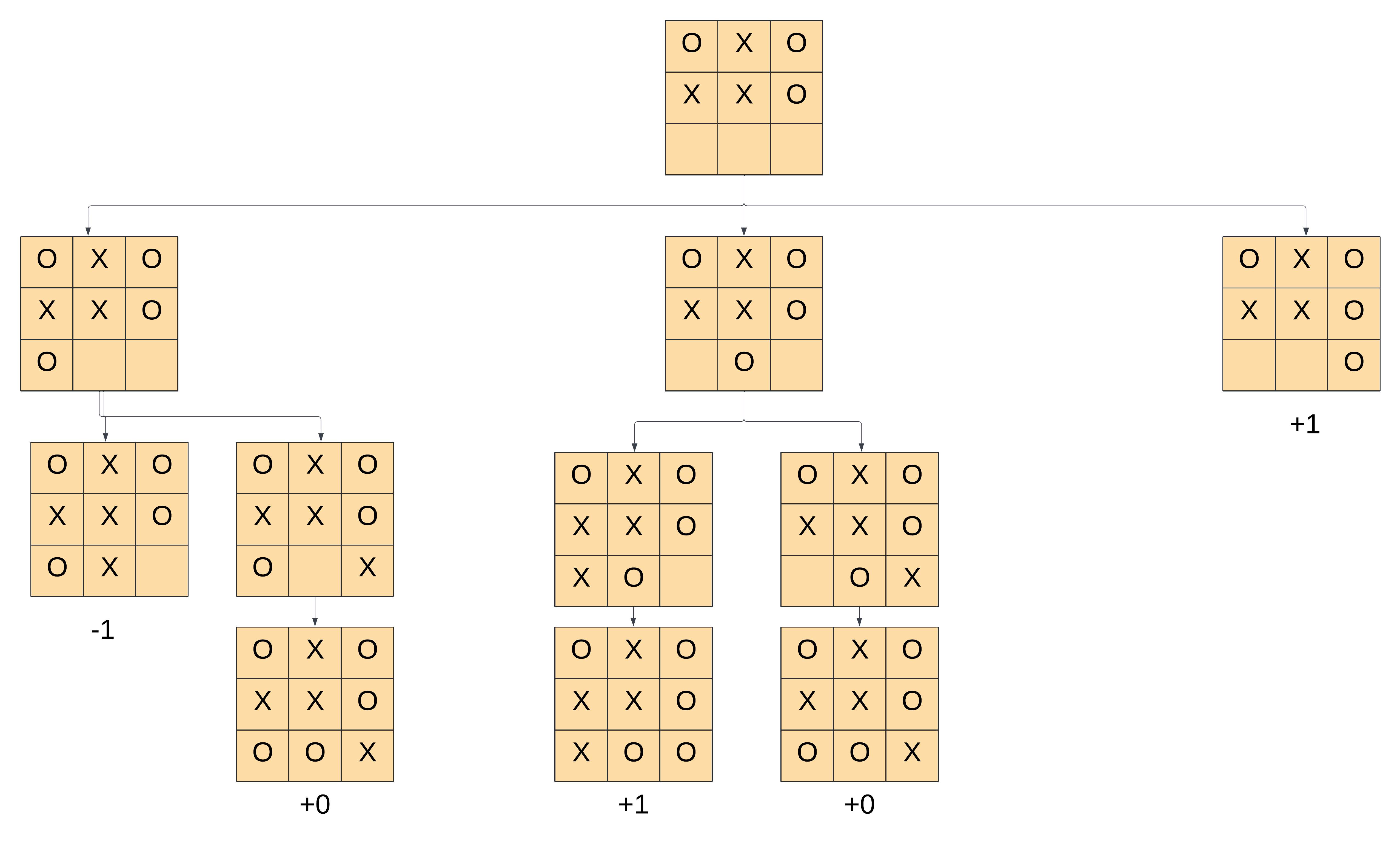
**.** Win/loss/tie detection : game detects these states and provide appropriate messages

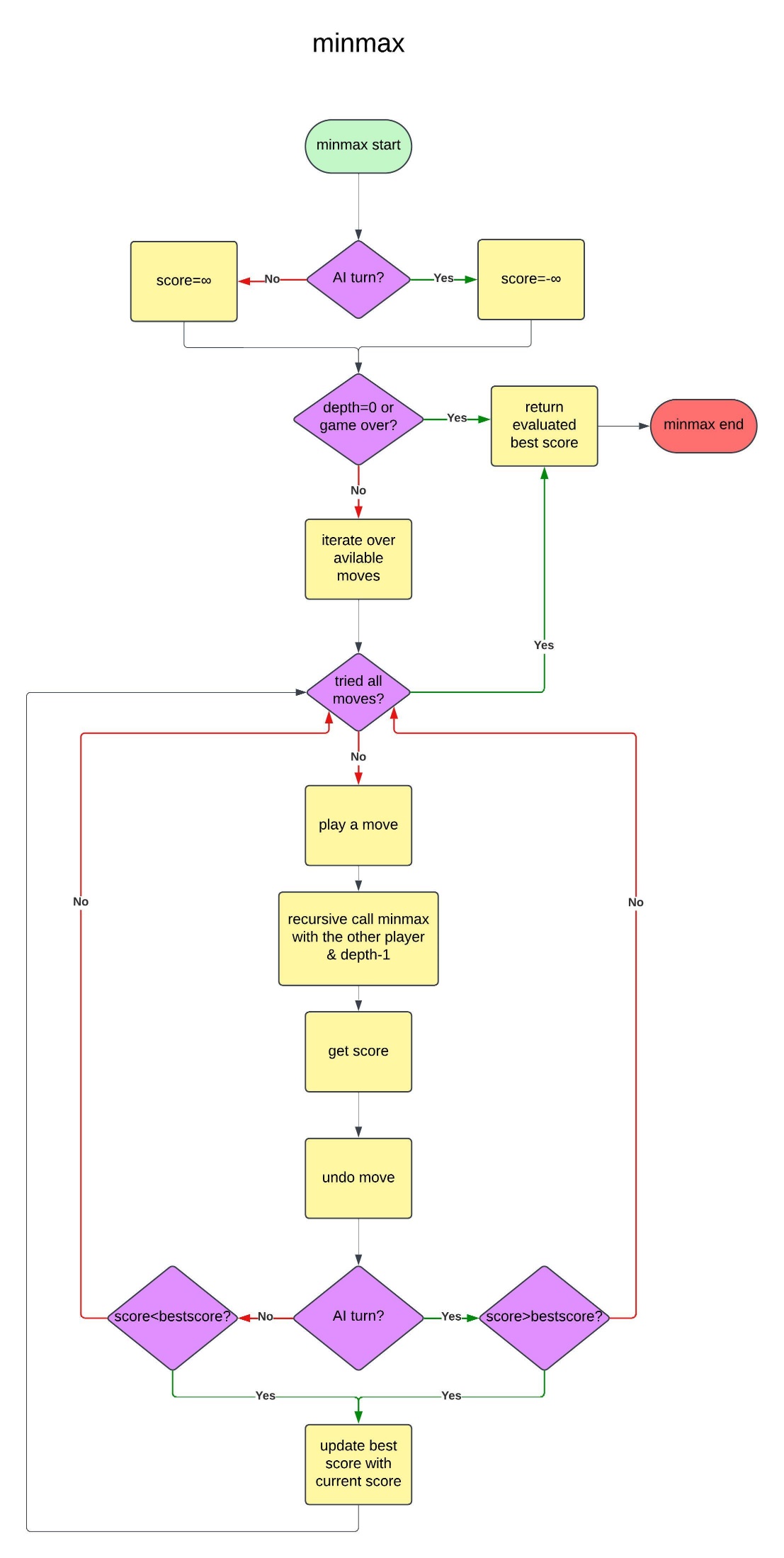
****

**Applied algorithms**

**1)Minmax**

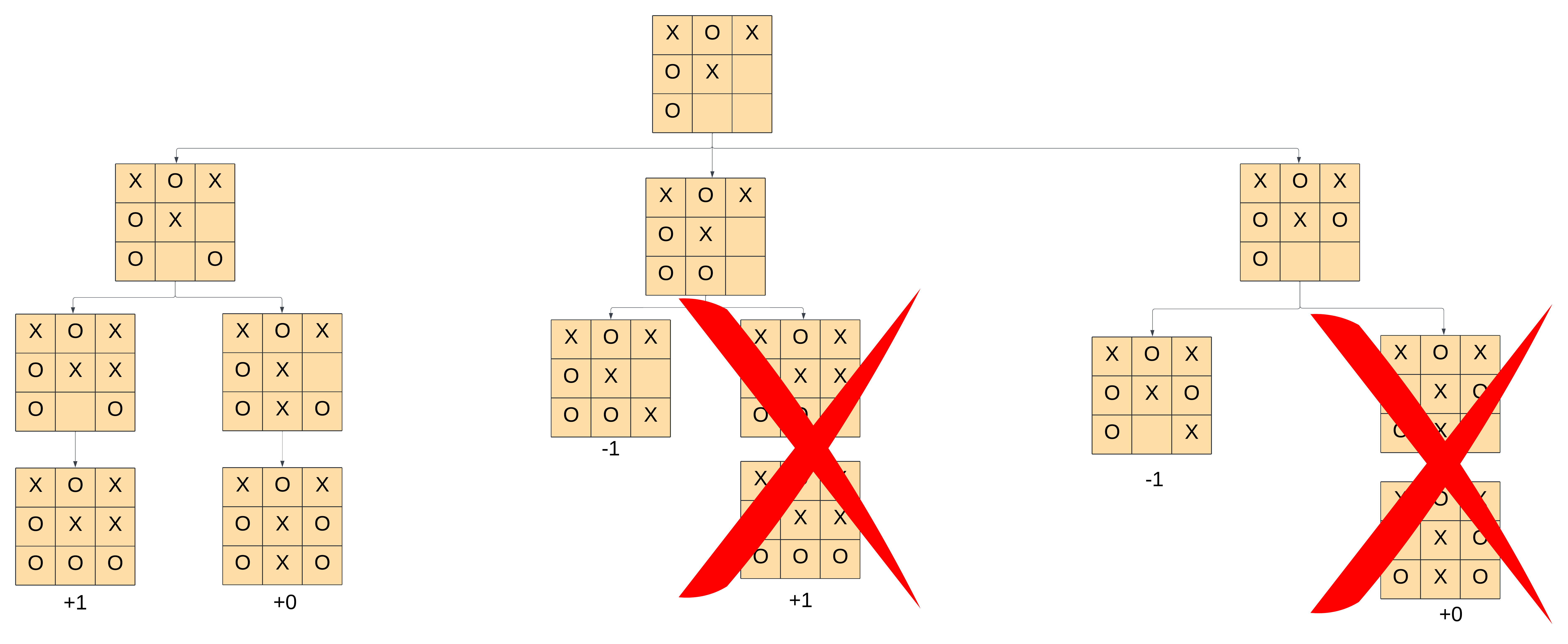
Explore all possible moves to find best move assuming the opponent plays optimally

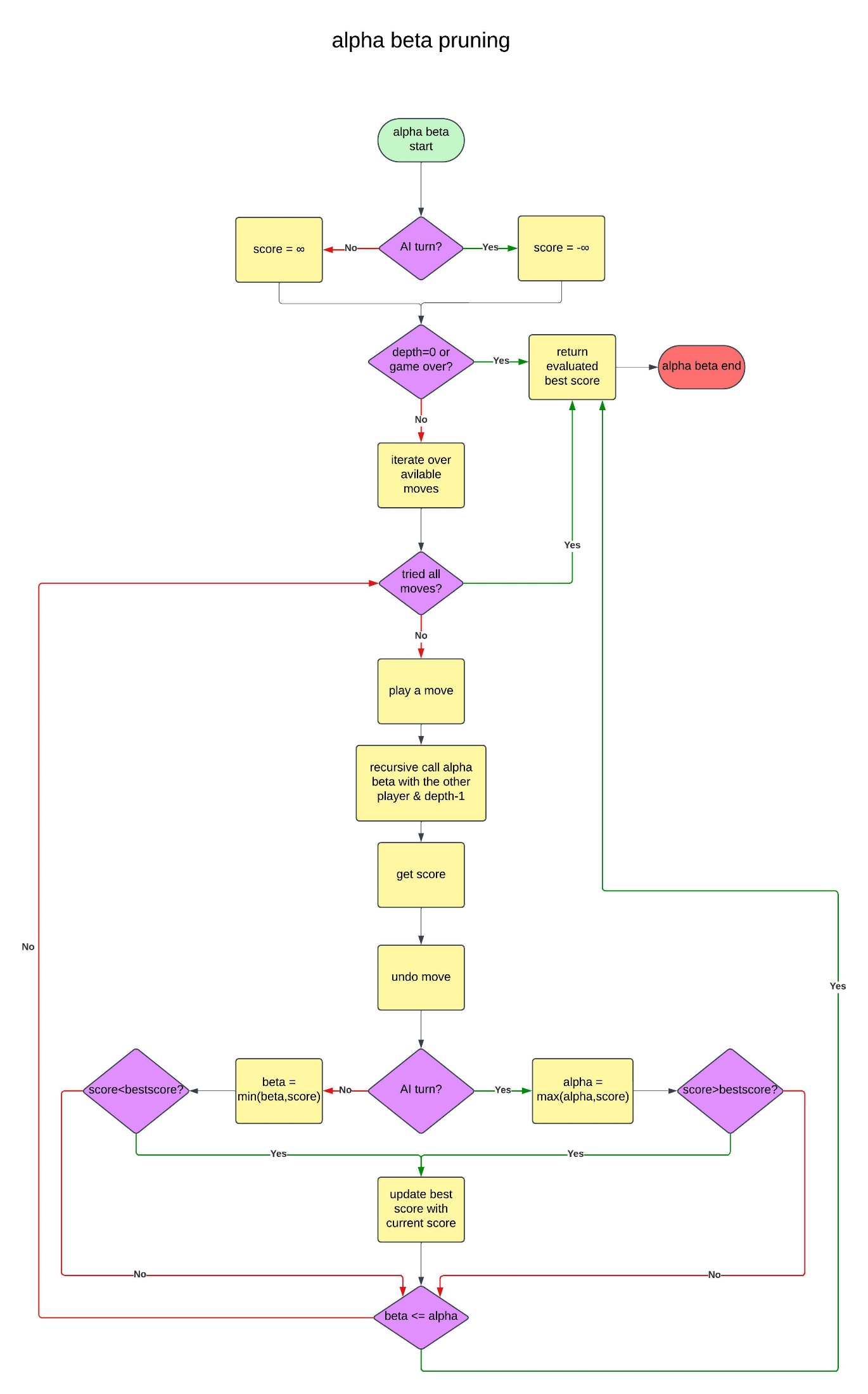
****

****

**2)Alpha beta pruning**

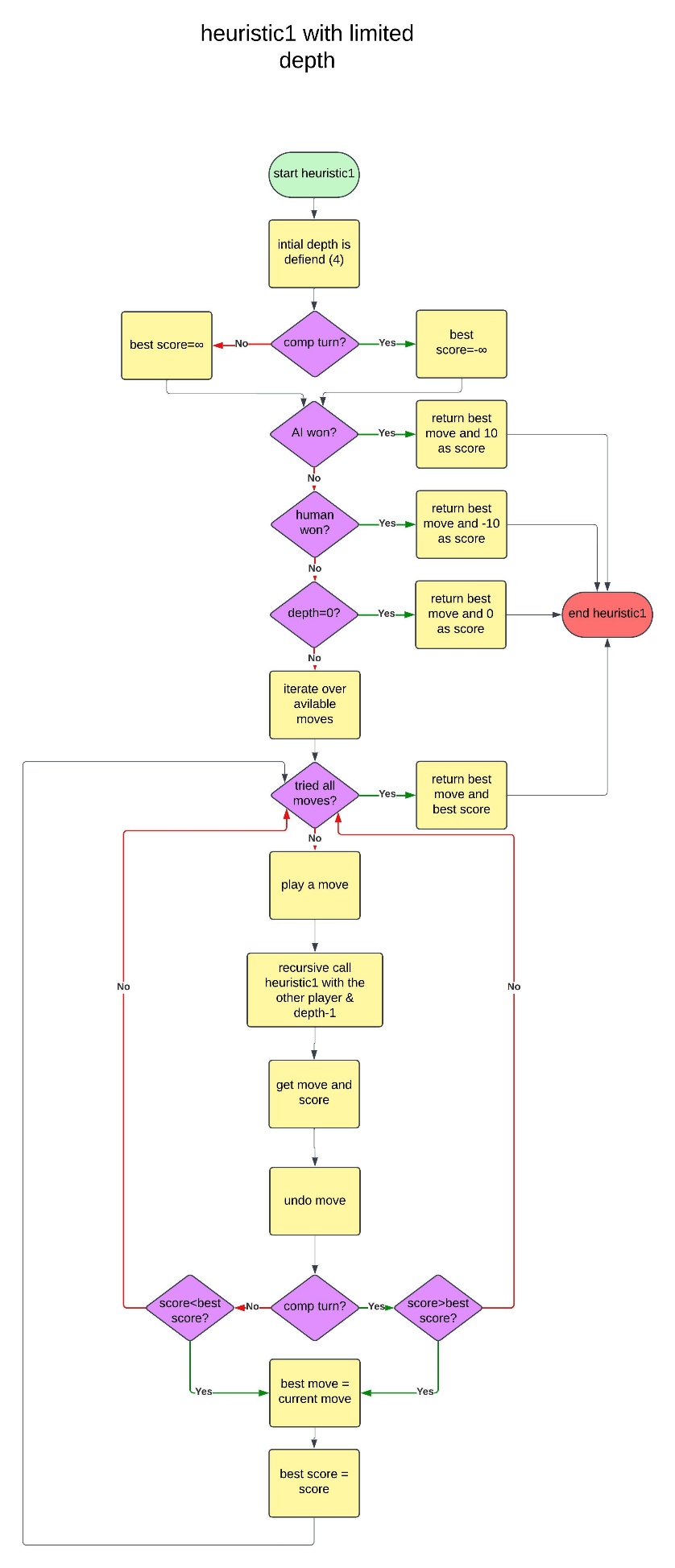
Optimize minmax by cutting off branches that won’t affect final decision

****

****

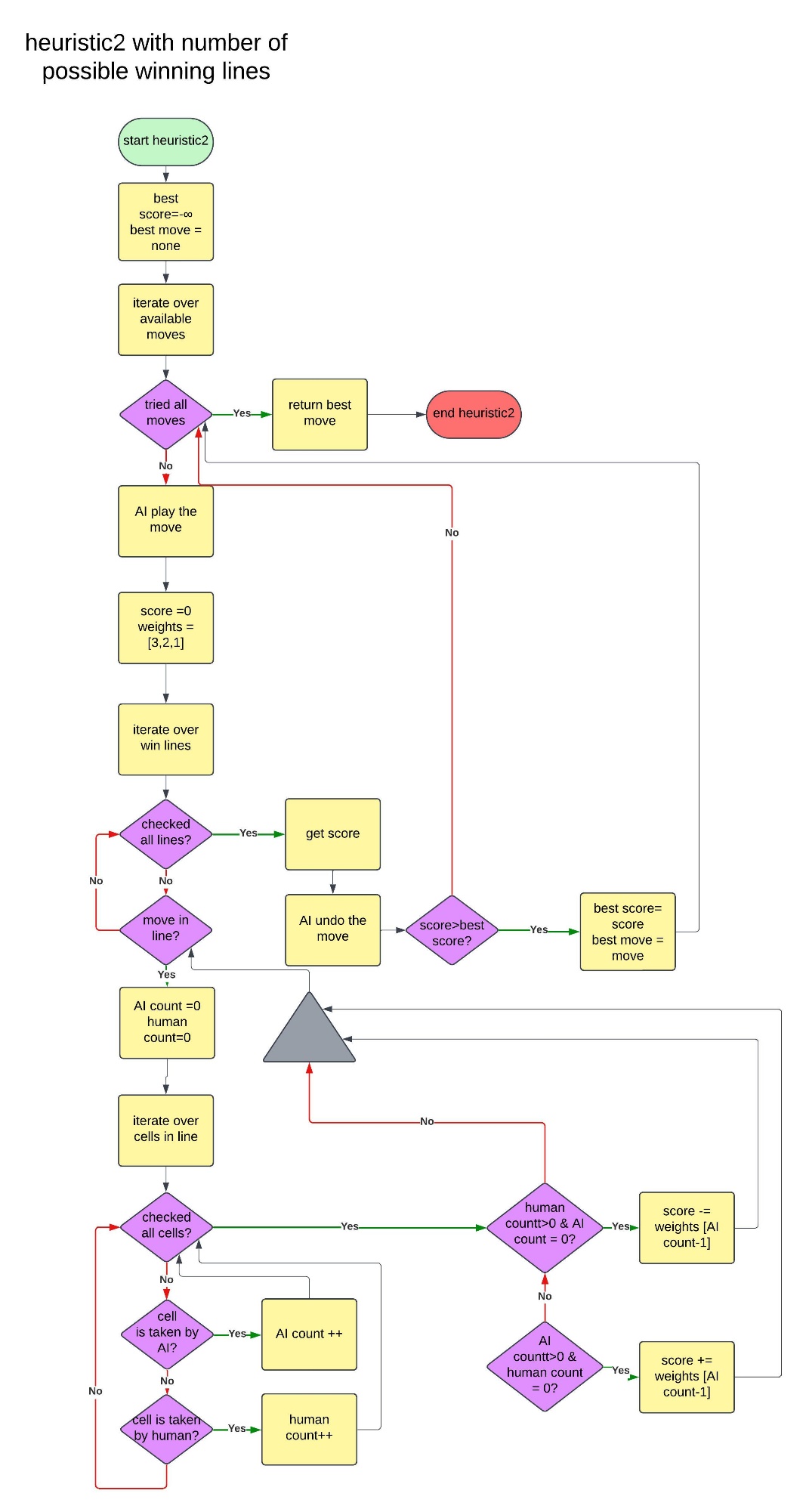
**3)First heuristic**

Approximate evaluation of non-terminal states to limit tree depth by using a fixed depth

****

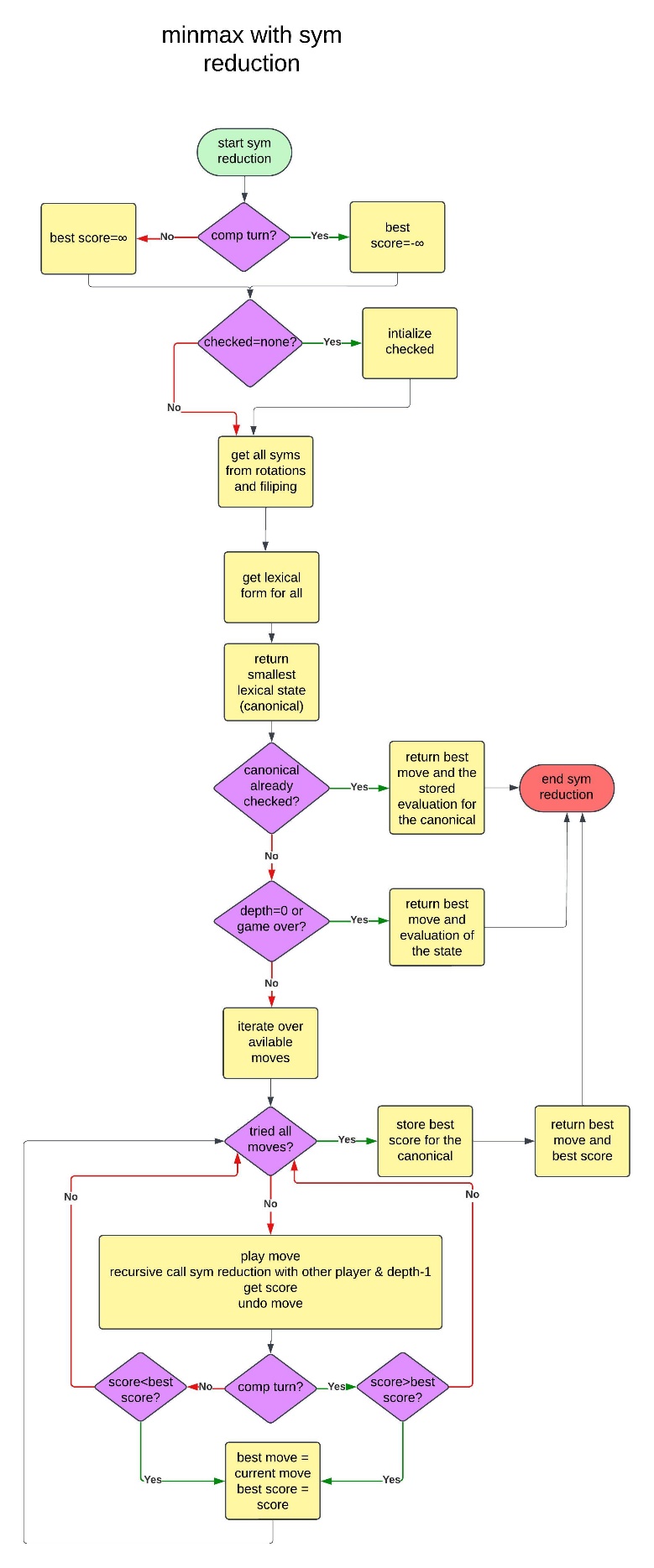
**4)Second heuristic**

Quick evaluation based on the potential winning lines

****

**5)Symmetry reduction**

Reduces redundant evaluation by considering only unique board states using rotates and flips it generates all symmetric variations and evaluates the canonical form

****

**Experiments and results**

**Testing**

Verified each algorithm’s correctness by playing multiple games

**Results**

**Minmax:** guaranteed optimal results but expensive for large depth

**Alpha beta pruning:** faster than minmax without loss of optimality

**Heuristic 1:** faster with acceptable decision quality

**Heuristic 2:** best performance for casual play but occasionally suboptimal

**Symmetry reduction:** reduced computational time in symmetry states

**Analysis, discussion and future work**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Time complexity** | **Space complexity** | **Optimality** |
| **Minmax** | **O(b^d)** | **O(d)** | **Always optimal** |
| **Alpha beta** | **O(b^(d/2))** | **O(d)** | **Always optimal** |
| **First heuristic** | **O(b^dmax)** | **O(dmax)** | **Not necessarily optimal** |
| **Second heuristic** | **O(1)** | **O(1)** | **Not necessarily optimal** |

Where b= branching factor

d= depth

dmax= max depth that was decided in the function

**Advantages and disadvantages**

Minmax

Advantages

Guarantees optimal play

Simple to implement

Provides deterministic outcomes

Disadvantages

Expensive for large games

Requires large memory

Alpha beta pruning

Advantages

Guarantees optimal play like minmax

Reduces computational load by pruning unnecessary branches

Speed up decision making

Makes deeper search feasible

Disadvantages

More complex to implement compared to minmax

Efficiency depends on moves order

First heuristic

Advantages

Reduces computational load by setting a maximum depth

Speed up decision making

Easy to implement

Disadvantages

Doesn’t provide optimal solutions especially in edge cases

Second heuristic

Advantages

Very low computational load

Speed up decision making

Easy to implement

intuitive

Disadvantages

Doesn’t provide optimal solutions

Doesn’t account for opponent’s moves very well

Application

Advantages

Demonstrates AI techniques effectively

Provides a user friendly interface

Disadvantages

Limited scalability beyond 3x3 boards

Heuristics may not perform optimally in edge cases

Future work

Extend game to larger board size

Incorporate machine learning for adaptive difficulty

Add a feature for players to analyze AI's decision-making process

Optimize heuristics especially in edge states

Conclusion

This project combines classical AI techniques and user-friendly design to create a practical Tic-Tac-Toe application. It serves as a valuable learning resource for understanding game AI fundamentals