# CS404 Assignment3

Ozan Başaran 26805

May 2023

## 1 Question 1

**Players:**

Player 1: Human Player

- Symbol: '/'
- Description: Represents the human player who places the right diagonal line ('/') on the grid.

Player 2: AI Player

- Symbol: '\'
- Description: Represents the AI player who places the left diagonal line ('\') on the grid.

**States:**

A state in the game consists of the following components:

- Grid: A square grid of size $m \times m$, where each cell can be empty or occupied by a player's symbol ('/' or '\').
- Intersections: A set of intersections on the grid, where each intersection contains a circle with a number ranging from 0 (used to denote '*') to 3.
- Player Scores: The current scores of both players, indicating the points they have accumulated throughout the game.

**Initial State:**

The initial state of the game is defined as follows:

- Grid: An empty grid of size $m \times m$, where each cell is initially unoccupied.
- Intersections: A set of intersections on the grid, where each intersection contains a circle with a number ranging from 1 to 3.
- Player Scores: Both players start with a score of 0.

At the beginning of the game, the grid is empty, and the intersections contain circles with their respective numbers. The players' scores are set to 0, indicating that no points have been accumulated yet. The game is ready to start with the first player making a move.

**Terminal States:** The game ends when all cells on the grid are filled. At this point, the game enters a terminal state. The payoff function is applied to the final state to determine the winner.

**State Transition Function:**

Given the current state of the game, including the grid configuration and the player whose turn it is, the state transition function updates the game state by placing the player's symbol on a specific cell of the grid and toggling the turn to the next player.

It can be defined as follows:

$$transition(currentplayer, currentgrid) \rightarrow nextplayer, nextgrid$$

The updated grid and the next player are returned as the output of the state transition function.

**Payoff Function and AI Decision-Making:**

During the game, players can earn points based on the number of diagonal lines that intersect at a circle on the grid. Each circle contains a number ranging from 1 to 3. If the number of intersecting lines at a circle is equal to the number on that circle, and no player has received points from that circle before, the player gets points. For every n points earned, the opponent receives -n points. It is important to note that a player can earn points from a circle at most once.

The AI's decision-making process is based on the minimax algorithm, which aims to maximize its expected score while considering the opponent's moves. The AI is the maximizing player, seeking to make moves that maximize its score, while Player 1 (human player) is the minimizing player, aiming to make moves that minimize the AI's score.

To optimize the decision-making process, we employ alpha-beta pruning, which reduces the number of nodes explored in the game tree. Alpha-beta pruning eliminates branches that are guaranteed to be worse than previously evaluated moves, significantly improving the efficiency of the search algorithm. By pruning unnecessary branches, the AI can quickly identify the best move without examining all possible moves exhaustively.

The AI evaluates the potential moves by simulating the game for each possible move, considering the resulting game states and the scores obtained. The minimax algorithm, along with alpha-beta pruning, helps the AI make informed decisions by considering the future consequences of each move.