# CAP 4053 Artificial Intelligence for Computer Games: Game Trees

# CAP 4053 Artificial Intelligence for Computer Games: Lecture Notes on Game Trees

## Introduction

Game trees are a type of data structure used to represent all possible moves in a game. They are used to model the decision-making process of a player or AI agent in a game. Game trees are composed of nodes and branches, where each node represents a game state and each branch represents a move. By traversing the tree, we can determine the best move to make in any given situation.

## Key Concepts

- **Node**: A node is a point in the game tree that represents a game state. It contains information about the current board position, the players' pieces, and the available moves.

- **Branch**: A branch is a line connecting two nodes in the game tree. It represents a move from one game state to another.

- **Root Node**: The root node is the starting point of the game tree. It represents the initial game state.

- **Leaf Node**: A leaf node is a node at the end of a branch. It represents a terminal game state, such as a win or a draw.

- **Minimax Algorithm**: The minimax algorithm is an AI technique used to determine the best move in a two-player game. It evaluates all possible moves and chooses the one that leads to the best outcome.

## Coding Examples

### Example 1: Creating a Game Tree

```
Start of Code
// Create a root node
let rootNode = {
        boardPosition: [], // array of pieces
        availableMoves: [], // array of possible moves
        children: [] // array of child nodes
};
// Create a child node
let childNode = {
        boardPosition: [], // array of pieces
        availableMoves: [], // array of possible moves
        parent: rootNode // reference to parent node
};
// Add the child node to the root node
rootNode.children.push(childNode);
```

End of Code

### Example 2: Traversing a Game Tree

Start of Code

```
// Traverse the game tree using the minimax algorithm
function traverseTree(node) {
        // Base case: if the node is a leaf node, return its value
        if (node.children.length === 0) {
                return node.value;
        }
        // Recursive case: evaluate the children of the node
        let bestValue = -Infinity;
        for (let child of node.children) {
                let value = traverseTree(child);
                bestValue = Math.max(bestValue, value);
        }
        return bestValue;
}
```

End of Code

## Practice Questions

1. What is a node in a game tree?

A. A node is a point in the game tree that represents a game state. It contains information about the current board position, the players' pieces, and the available moves.

2. What is the minimax algorithm?

A. The minimax algorithm is an AI technique used to determine the best move in a two-player game. It evaluates all possible moves and chooses the one that leads to the best outcome.