

# **Minimax and Alpha/Beta Pruning for Tic-Tac-Toe**

Frederik Blem - [cph-fb114@cphbusiness.dk](mailto:cph-fb114@cphbusiness.dk)  
Simon Schønberg Bojesen - [cph-sb339@cphbusiness.dk](mailto:cph-sb339@cphbusiness.dk)

May 3, 2021

Copenhagen Business Academy  
Supervisor: Martin Vestergaard

## **Abstract**

This is where an abstract will be in 4 sentences! What is the problem? Why is the problem interesting? What is the solution? What are the implications of the solution?

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Minimax Algorithm</b>	<b>4</b>
2.1	Introduction to Minimax . . . . .	4
2.2	Explanation of Minimax . . . . .	4
2.3	Implementation of Minimax . . . . .	4
<b>3</b>	<b>Alpha/Beta Pruning</b>	<b>5</b>
3.1	Introduction to Alpha/Beta Pruning . . . . .	5
3.2	Explanation of Alpha/Beta Pruning . . . . .	5
3.3	Implementation of Alpha/Beta Pruning . . . . .	5
<b>4</b>	<b>Conclusion</b>	<b>6</b>

## List of Figures

## List of Tables

# 1 Introduction

This paper is about the Minimax algorithm and Alpha/Beta pruning, which make up the core of our former assignment "Tic-Tac-Toe using Minimax" for the subject Data Science. We chose to cover this project over other projects because the topics were among the more interesting and we wanted to research these further.

One of the objective of this paper would be to shed some light on the Minimax Algorithm, what it is good for, why this is interesting and different ways to implement it. Another objective is showing how to optimize this algorithm further by implementing Alpha/Beta pruning.

We will do this by describing the logic behind the topics and showing implementations from the Tic-Tac-Toe assignment, that we will be modifying for the purpose of this paper.

## **2 Minimax Algorithm**

### **2.1 Introduction to Minimax**

The Minimax algorithm is a recursive algorithm which is used in turn-based games like chess and tic-tac-toe, to make the AI player consider all possible moves and choose the best one.

### **2.2 Explanation of Minimax**

### **2.3 Implementation of Minimax**

## **3 Alpha/Beta Pruning**

### **3.1 Introduction to Alpha/Beta Pruning**

Alpha/Beta pruning is the way by which the Minimax Algorithm is made more efficient. This happens by stopping consideration of possible moves, that will never be better than previously considered moves.

### **3.2 Explanation of Alpha/Beta Pruning**

### **3.3 Implementation of Alpha/Beta Pruning**

## 4 Conclusion