Licenciatura em Engenharia Informática

Unidade Curricular de

**Programação Orientado a Objetos**

  
 (2º ano/2semestre – 2019 / 2020)

**TIC TAC TOE**

**Relatório Técnico – Mini Projeto**

*Docentes*:

José Valente de Oliveira

*Discentes*:

PL1 Grupo nº5

Guilherme Correia, nº61098

Henrique Cruz, nº61099

Bruno Susana, nº61024

## Resumo

## Introdução

Este trabalho tem como objetivo aprofundar os nossos conhecimentos sobre inteligência artificial, implementando o algoritmo Monte Carlo Tree Search (MCTS) ao jogo do galo. Este jogo envolve dois jogadores e é formado por um tabuleiro 3 por 3, tem como objetivo colocar três peças iguais em linha, quer horizontal, vertical ou diagonal.

Devido à simplicidade do tic tac toe, é frequentemente usado como uma ferramenta [pedagógica](https://en.wikipedia.org/wiki/Pedagogical) para ensinar os conceitos de bom [desportivismo](https://en.wikipedia.org/wiki/Sportsmanship) e o ramo da [inteligência artificial](https://en.wikipedia.org/wiki/Artificial_intelligence) que trata da busca em [árvores](https://en.wikipedia.org/wiki/Game_tree) de [jogo](https://en.wikipedia.org/wiki/Game_tree) . É simples escrever um [programa de computador](https://en.wikipedia.org/wiki/Computer_program) para jogar o tic tac toe perfeitamente ou enumerar as 765 posições essencialmente diferentes ou os 26.830 jogos possíveis [até](https://en.wikipedia.org/wiki/Up_to) rotações e reflexões neste espaço. Se jogado de forma ideal por ambos os jogadores, o jogo termina sempre em empate, tornando o jogo do galo um [jogo fútil](https://en.wikipedia.org/wiki/Futile_game)



Figura 1-jogo tic tac toe

## Designs implementados

O 2º design implementado foi atribuir ao numero de vitorias valores no intervalo [0,1], inicialmente foi usando uma técnica que é usando no algoritmo minimax, que consiste em subtrair ao valor da vitoria -1 por cada jogada necessária realizar, tanto nossa quanto do inimigo, desta forma a tornar o programa mais inteligente no momento de decidir qual jogada devera fazer. Esta forma não teve sucesso porque os resultados que se obtida não era os esperados.

O 3º e 4º design implementados são muitos parecidos, ambos contam as vitorias de forma igual mas na fase de backpropagation, a distribuição dos valores é que diferencia, no caso do design 3 ele retro propagava -1 em caso de derrota e em caso de empate retro propagava 0.5 o que não fazia sentido