

PLOG 2019/2020 – Trabalho 1

Asterismo

Relatório Intercalar

(20 de outubro de 2019)



José António Barbosa Fonseca Guerra **up201706421**@fe.up.pt

Martim Pinto da Silva **up201705205**@fe.up.pt

# Índice

* Introdução ………………………………………………………. 3
* Jogo Asterismo ………………………………………………… 4
* Lógica do Jogo …………………………………………………. 6
  + Representação interna do estado do jogo …. 6
  + Visualização do tabuleiro ………………………….. 9
  + Lista de jogadas Válidas ……………………………. 11
  + Execução de jogadas………………………………….
  + Final do Jogo………………………………………………
  + Avaliação do Tabuleiro…………………………….
  + Jogada do Computador……………………………
* Conclusões …………………………………………………….
* Bibliografia

# Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de Programação em Logica (PLOG), do 3º ano do curso de Engenharia Informática da FEUP.

O objetivo deste projeto foi implementar, através do sistema de desenvolvimento do Prolog, um jogo de tabuleiro com um conjunto de regras definidas pelo criador do jogo.

O relatório segue a seguinte estrutura:

* Jogo Asterismo
* Lógica do Jogo
  + Representação interna do estado do jogo
  + Visualização do tabuleiro
  + Lista de jogadas válidas
  + Execução de jogadas
  + Final do jogo
  + Avaliação do tabuleiro
  + Jogada do computador
* Conclusões
* Bibliografia

# Jogo Asterismo

# 

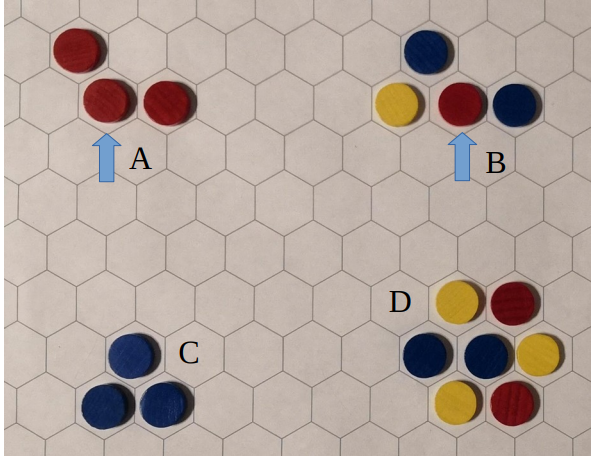
Este jogo de tabuleiro foi criado em 2019 por Giuliano Polverari.

É um **jogo cooperativo** para ser jogado entre 2 pessoas jogado num tabuleiro com peças hexagonais.

Há 63 peças de 3 cores diferentes (vermelho, amarelo e azul).

O objetivo de cada jogador é obter 5 peças de cada cor, **sem perturbar o equilíbrio da arvore de jogo.**

É um jogo cooperativo, no qual os 2 jogadores jogam em equipa por um objetivo em comum: *acumular cada um 5 peças de cada cor.*



**Figura 2**: Explicação

Prémios:

* 2º lugar Cogita 2019 ("melhor jogo abstrato de estratégia”) 5ª edição.

Regras:

Qualquer peça pode ser retirada, no entanto as peças adjacentes têm de estar **seguras**.

Uma peça está **segura** se estiver ligada a duas peças da mesma cor (situação A) ou ligada a 3 peças de qualquer cor (situação B).

Cada jogador por turno faz apenas uma jogada, ou seja remove 1 peça, se possível.

Cada jogador tem um limite máximo de 5 peças para cada cor.

Ao retirarem peças, os jogadores têm de verificar se **não quebram a arvore**.

Se os 2 jogadores chegarem as 5 peças de cada cor vencem “o jogo”, senão “o jogo” vence e os jogadores saem derrotados.

**Figura 1**: tabuleiro do jogo



Situação inicial:

As peças são colocadas aleatoriamente pelo tabuleiro.

As peças, que devido a configuração inicial não estejam **seguras**, terão de ser atribuídas a cada jogador da forma aleatória em igual número.

**Figura 3**: Situação inicial

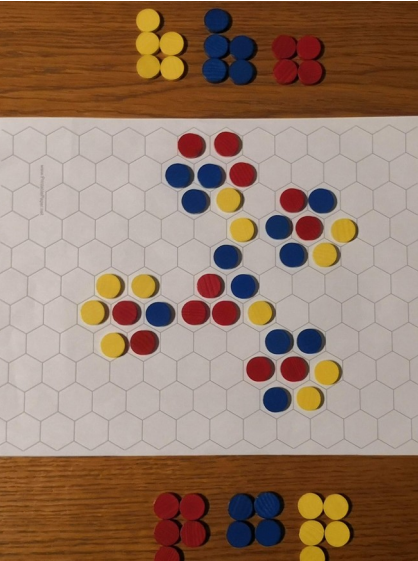


Desenvolvimento:

Vai-se retirando e colocando peças, respeitando as regras de jogo, tendo como objetivo principal acumular 5 peças de cada cor.

Os 2 jogadores jogam cooperativamente, ou seja, têm de acumular as dois 5 peças de cada cor para alcançar a vitória.

O jogador terá de ter especial atenção para não quebrar a arvore do jogo, isto é não poderá haver blocos sem peças adjacentes.



**Figura 5**: Situação final

Situação final:

O jogador terá de estar atento a situações em que possui 14 peças, assim como o seu colega de equipa.

Nestas situações, é provável que não seja possível retirar mais peças sem quebrar a árvore do jogo. Nesta condição perdem os jogadores e ganha “o jogo”, tal como neste exemplo. Se por outro lado for possível cada 1 alcançar as 5 peças de cada cor os 2 jogadores vencem “o jogo”.

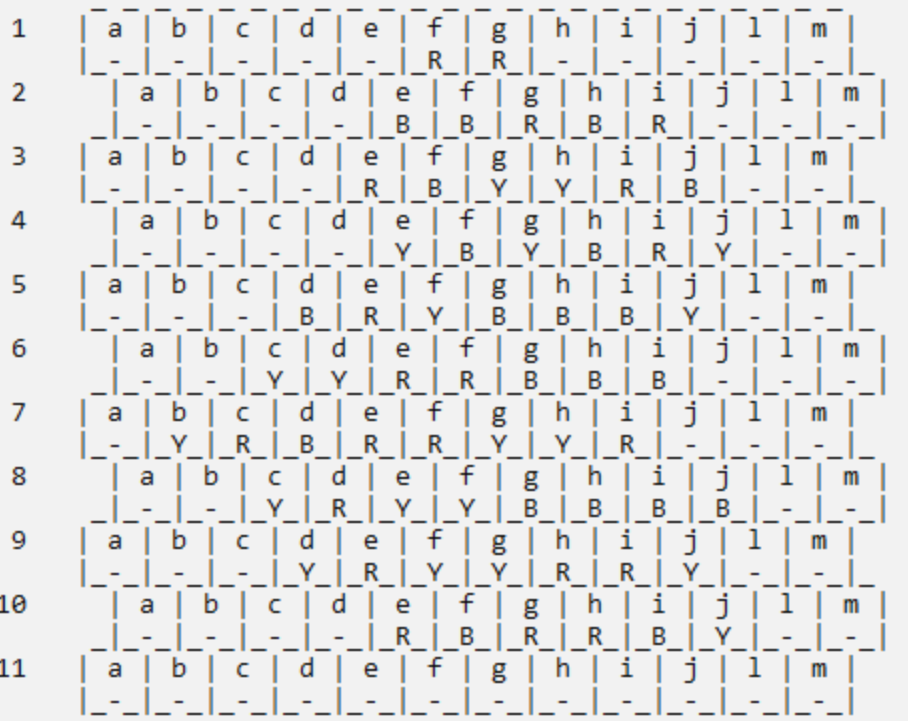
**Figura 4**: Situação desenvolvimento

# Lógica do Jogo

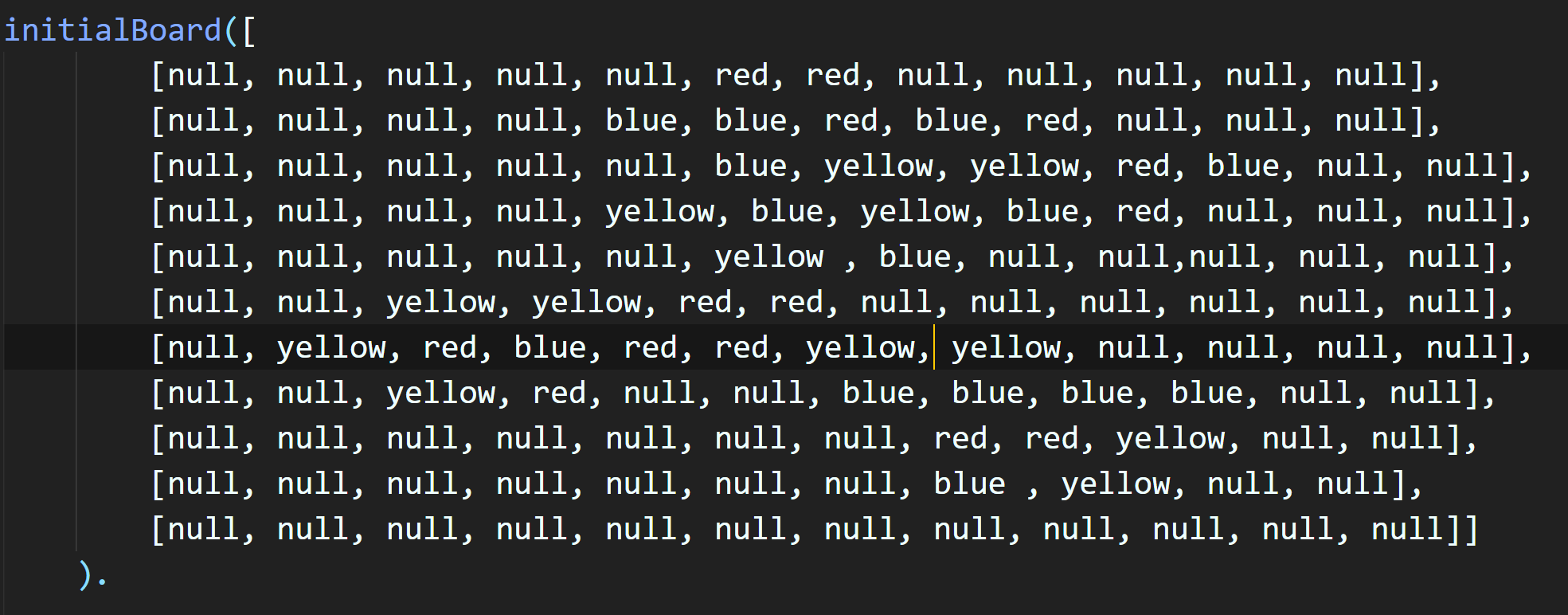
# Representação interno do estado do jogo

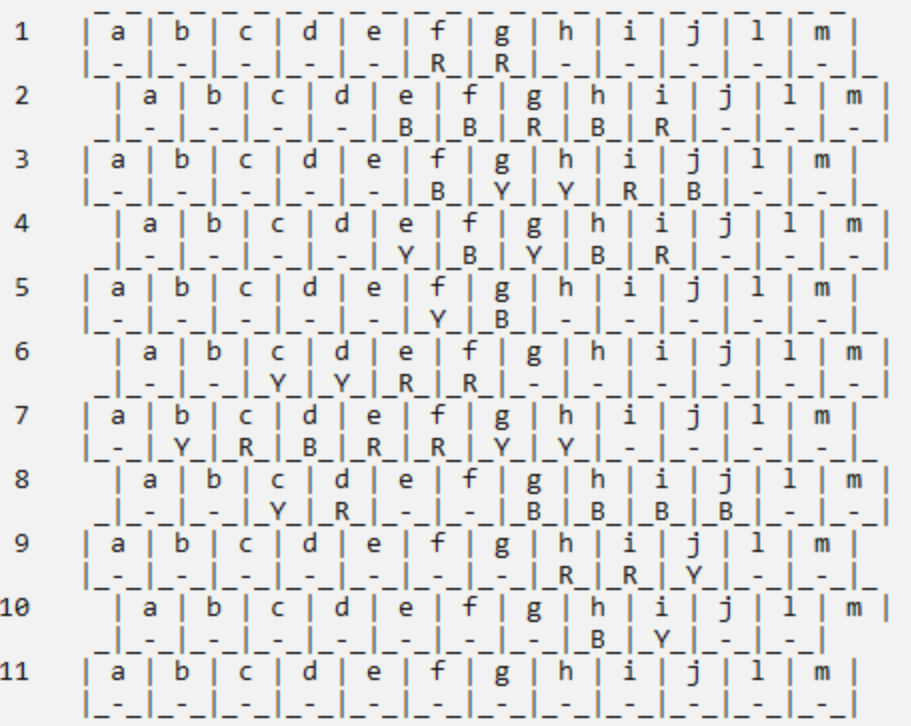
# Situação inicial:

# 



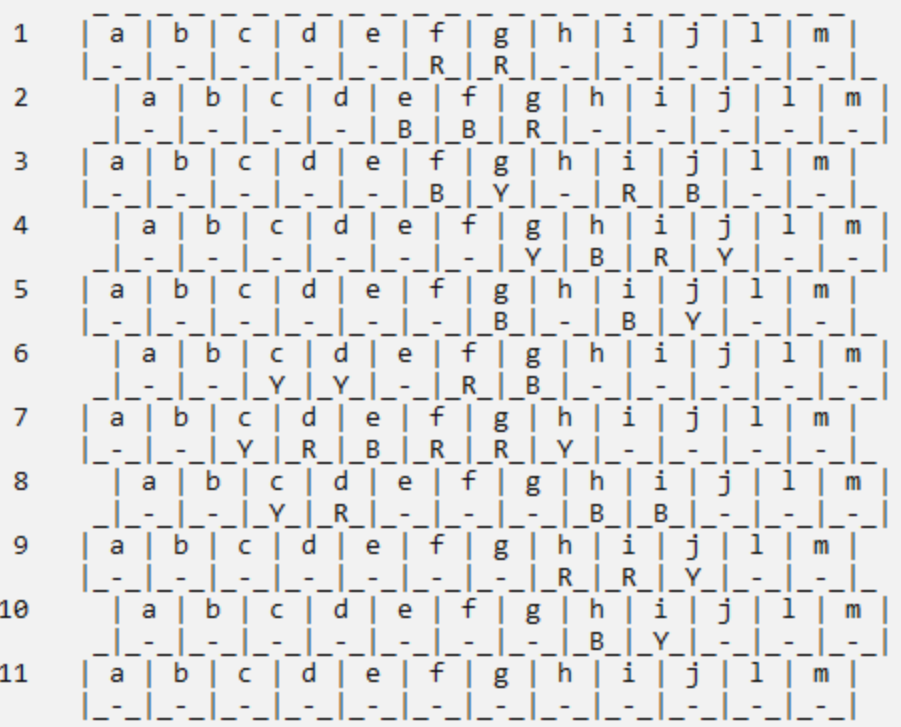
# Desenvolvimento:





# Situação final:

# 



# Visualização do tabuleiro – atualizar print

# 

# Predicados

# Cabeçalho do predicado para adicionar e remover uma peça (remover no sentido de colocar um ‘-‘ no espaço que ficou vazio)

# addPiece(InitialBoard, Color, Row, Column, NewBoard).