

#### Inteligência Artificial

#### DR. JOSÉ VALENTE DE OLIVEIRA

# Relatório

## Problema N-queens

### Grupo IA2223PL2G04:

#### Diogo Jorge, a57593

#### Gonçalo Almeida, a71262

#### Tomás Roma, a71269

# **Índice**

## **Descrição do problema**

## **Design**

## **UML de implementação**

## **Análise de resultados**

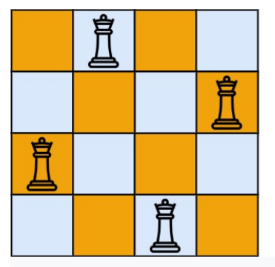
## **Conclusões**

## Descrição do problema e algoritmos usados

O problema N-queens é um problema clássico de xadrez que consiste em posicionar N rainhas dentro de um tabuleiro de xadrez sem que nenhum par de rainhas se ataquem. Este tabuleiro tem um tamanho NxN que não se altera.

O input é um valor inteiro N que representa o número de queens que serão colocadas no tabuleiro.

O output esperado será uma matriz NxN com todas as rainhas posicionadas corretamente.

Para a resolvermos o problema, experimentamos implementar diferentes estratégias/algoritmos sendo estes: A\*, BestFirst. Para além destes algoritmos usamos também uma heurística complementar baseada nos conflitos presentes nos tabuleiros.

**Figura 1. Problema N-Queens com N = 4.**

Através da análise de resultados decidimos usar, o que pensamos ser mais eficiente, o algoritmo BestFirst.

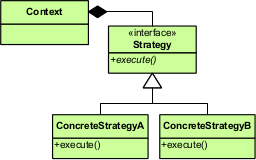
O BestFirst é um algoritmo informado, através de uma heurística complementar. Para achar uma solução, ele cria uma árvore composta por nós. Esses são chamados de state, que por si são formados por objetos do contexto (que no nosso problema, são tabuleiros com diferentes disposições de rainhas). O algoritmo, BestFirst, interage com os nós e atua da seguinte forma:

1. Define o primeiro state (primeiro tabuleiro) criado como a “raiz” da árvore e posiciona-o na primeira posição de uma lista ordenada pelo número de conflitos de um state.
2. Pega no primeiro elemento da lista ordenada e vai gerar “filhos”, que serão novos tabuleiros que se definem pela troca de cada par de rainhas do state “pai”.
3. Coloca os filhos gerados na lista ordenada.
4. Repete os dois últimos passos até achar uma solução.

A heurística utilizada assume o número de conflitos (ataques entre rainhas) dentro do tabuleiro como o peso do mesmo. Ou seja, por exemplo se um tabuleiro contiver 5 conflitos então a nossa implementação vai assumir que g = 5, sendo g o peso do tabuleiro. Este será usado a fim de ordenar uma lista de elementos onde os estados/tabuleiros com menor g terão prioridade de expansão sobre os restantes.

Resumindo, a nossa implementação está a resolver o problema proposto através do algoritmo BestFirst e uma heurística complementar á base dos conflitos presentes no tabuleiro.

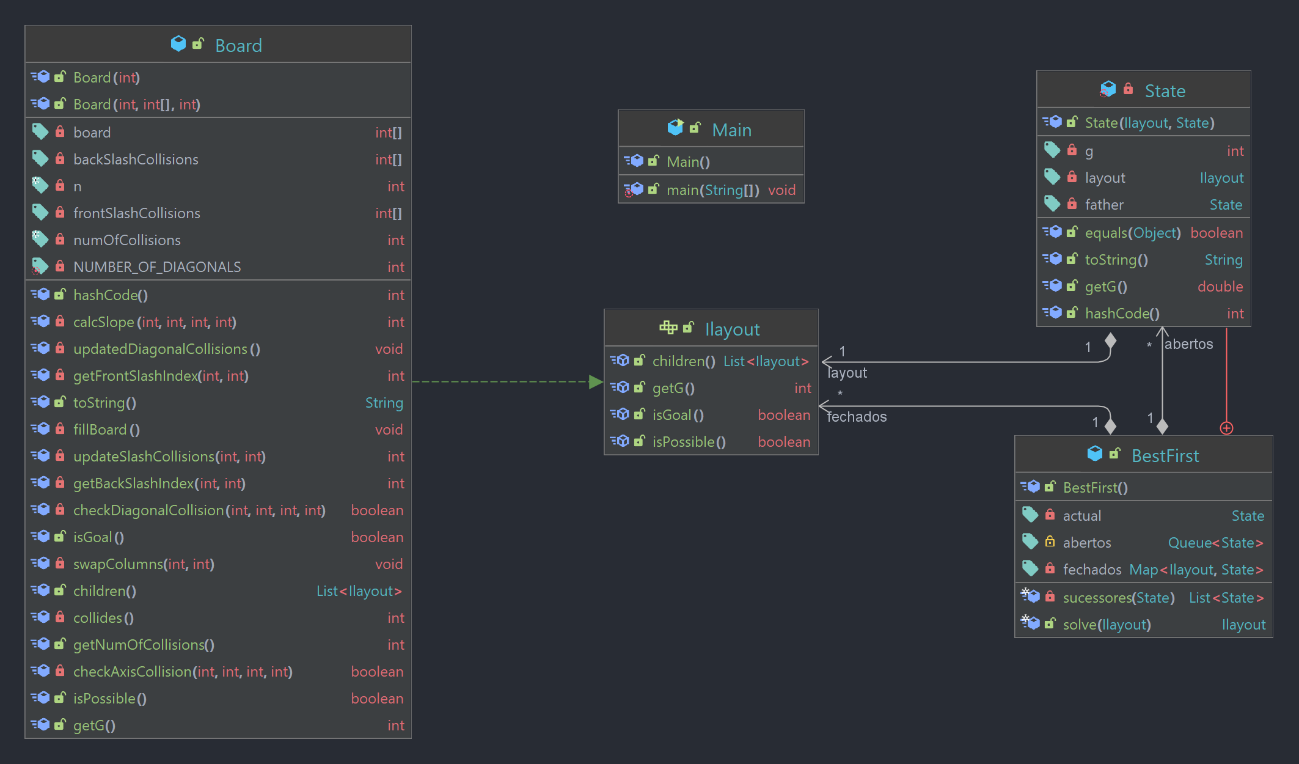
## Design



A nível de design utilizou-se o strategy design-pattern, dada a natureza do

problema que é possível ser resolvido com diferentes estratégias. A interface Ilayout foi usada como conector entre uma estratégia e o contexto. Neste projeto, o contexto é o problema N-Queens, e a estratégia/algortimo que utilizámos foi a BestFirst.

## UML de implementação



## Analise de Resultados

Ao realizar diferentes testes com vários valores para N, concluímos que para determinados casos o tabuleiro gerado inicialmente é o final, tornando a execução do programa O(n), dado que a complexidade temporal da função que gera o tabuleiro inicial é O(n). Os casos referidos são (relativos aos nossos testes) potências de 10 e de 4.

Para os restantes casos, a complexidade temporal é O(n2), que é a complexidade do algoritmo BestFirst.

O primeiro tabuleiro que não consegue ser resolvido é o n=698, uma vez que o número de estados guardados nas listas de abertos e fechados excede a memória, no entanto o algoritmo consegue resolver alguns tabuleiros maiores quando a geração de filhos leva-nos a um caminho mais fácil, como quando n = 700.

## Conclusões

Durante a implementação do projeto tivemos algumas dificuldades a otimizar o programa de forma a poupar memória não gerando nós pouco promissores.

Além disso, tivemos dificuldade a achar uma boa heurística para o algoritmo A\*, daí termos usado o BestFirst que achamos mais simples de implementar a heurística.

## Bibliografia

Slides disponibilizados pelo professor.

A polynomial time algorithm for the N-Queens Problem - <http://fizyka.umk.pl/~milosz/AlgIILab/10.1.1.57.4685.pdf>