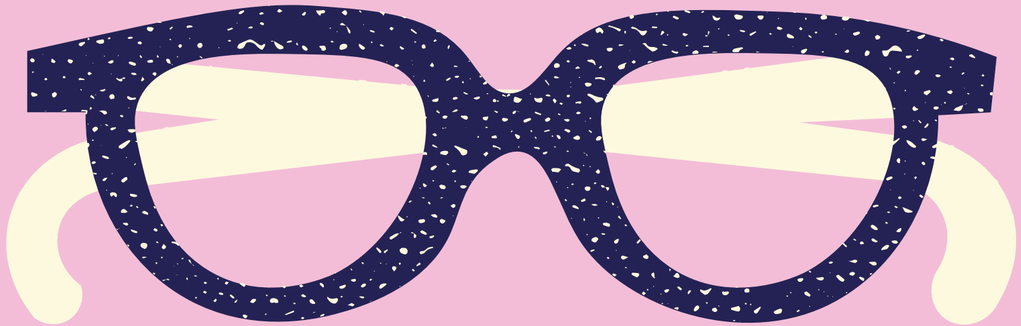
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Chess Snake puzzles

Heuristic Search Methods for One Player Solitaire Games

FEUP - LEIC - Artificial Inteligence 21/22

Luísa Marques - 201907565
Margarida Raposo - 201906784
Tomás Fidalgo - 201906743



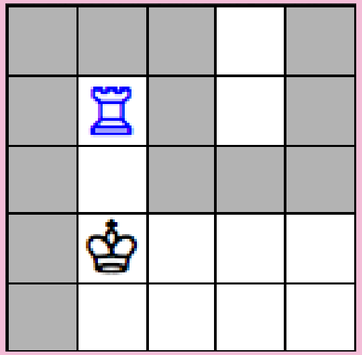
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Problem Specification

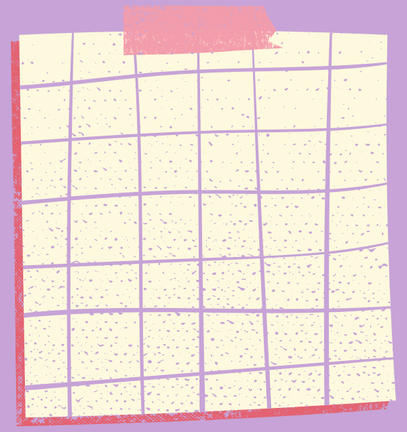
Problem State:

Given a board with chess pieces, draw a snake that goes from the lower left to the upper right square. The snake path moves horizontally or vertically and does not touch it self, even diagonally.
Given each chess piece possible moves, they should be able to attack the same number of segments of the snake.
Each puzzle has a unique solution, that can be calculated by getting all possible snake paths and after calculating the attack points of the chess pieces, selecting the correct path.

Solution representation:



U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



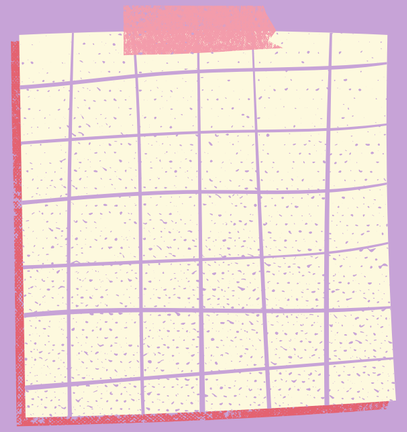
Formulation of the problem as a search problem

State Representation:
Position:[0, 1, X] (0 if the position is empty, 1 if the position belongs to the snake path, X for a letter representing a chess piece at the position (K -> king...))
CurrentPosition: (a,b)
Board: [[Position, Position,...][Position, Position,...]]

Initial State:
CurrentPosition: (0,0)
Board: [Position, Position,...] where every Position has a 0 or a piece

Objective State:
CurrentPosition: (X,X) where X is the length(Board)-1 and the Board is always a square
Board: [[Position, Position,...][Position, Position, ...]] where the first Position and the last one have 1

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



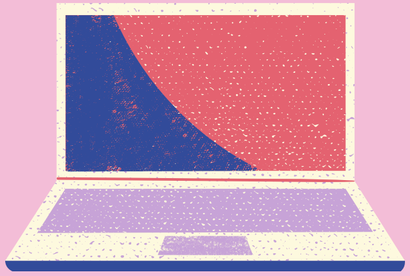
Formulation of the problem as a search problem

Operators:	MoveLeft()	MoveRigth()	MoveDown()	MoveUp()
PreCond	XCurPos < Length(Board)-1	XCurPos > 0	YCurPos > 0	YCurPos < Length(Board)-1
Effect	XCurPos = XCurPos - 1	XCurPos = XCurPos + 1	YCurPos = YCurPos - 1	YCurPos = YCurPos + 1
Cost	1	1	1	1

General PreConditions:

- A move can only be performed if all adjacent positions to the following one haven't been visited
- A move can only be performed if the following position hasn't been visited

Work already done



U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Programming language: Python

IDE: VSCode

Data Structures:

- Python Classes for different entities
- Lists for storing the problem domain and solution
- Graphs that represent the board

File Structure:

- src/ -> contains the source code for the implemented algorithms

Implemented:

- File Parsing