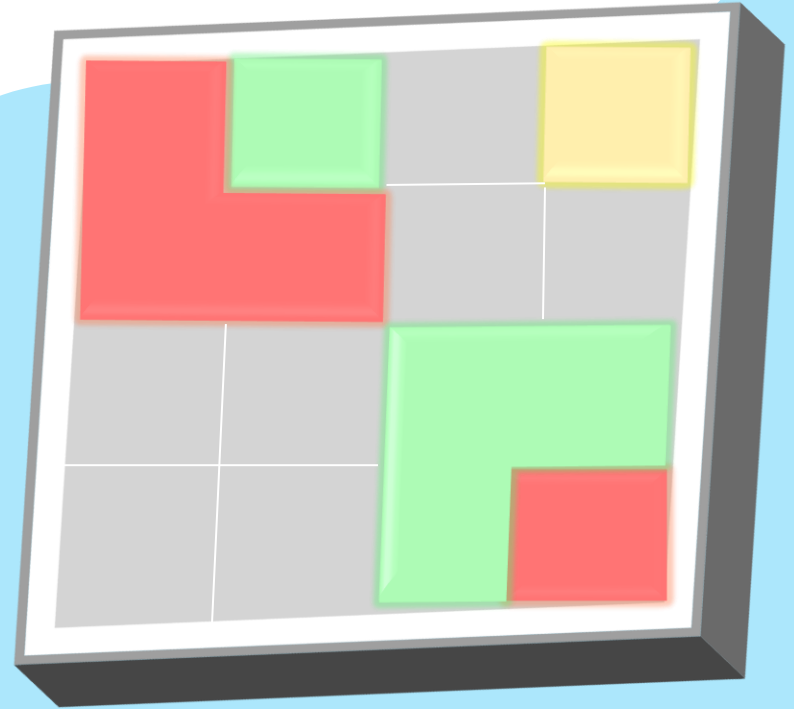


Cohesion

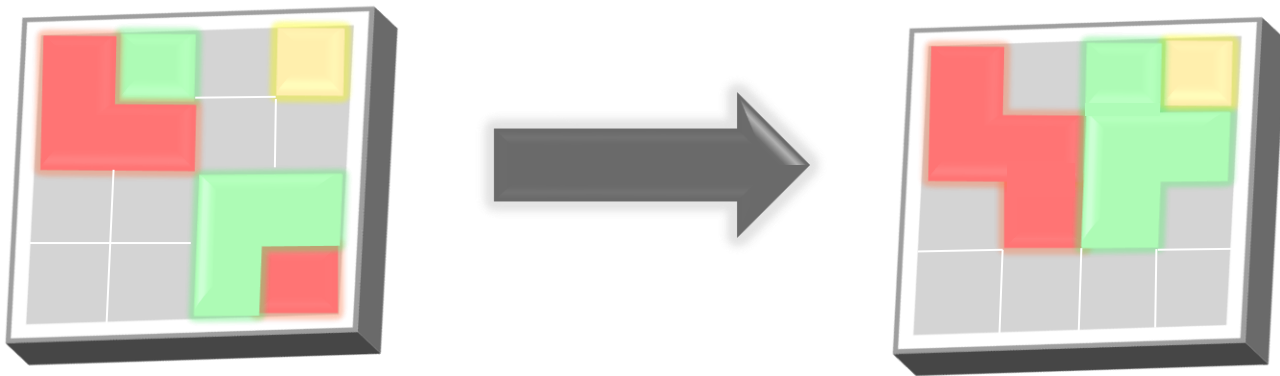
Artificial Intelligence
2022/2023 3LEIC10 Group 27

Daniela Tomás up202004946
Diogo Nunes up202007895
João Veloso up202005801



Problem Specification

Cohesion is a puzzle game where the player slides squares of different colors, and when two squares of the same color are placed next to each other, they bond together and become permanently attached. The puzzle is solved when all the squares of each color are connected.



Formulation of the problem

- **State Representation:** Matrix $M \times N$ composed of squares with different colors
- **Initial State:** Generated board and squares
- **Objective Test:** Check whether all the squares are connected to all other squares of the same color
- **Operators:**

Names	Preconditions	Effects	Costs
moveUp	There is an empty space above the square and all attached pieces of the same color will remain connected after the move	The square and all attached pieces move up by one position	1
moveDown	There is an empty space below the square and all attached pieces of the same color will remain connected after the move	The square and all attached pieces move down by one position	1
moveLeft	There is an empty space left to the square and all attached pieces of the same color will remain connected after the move	The square and all attached pieces move left by one position	1
moveRight	There is an empty space right to the square and all attached pieces of the same color will remain connected after the move	The square and all attached pieces move right by one position	1

Formulation of the problema (cont.)

- **Heuristics:**
 - **Heuristic 1 – Distance:**
 - **Heuristic 1.1** – Calculate the distance between all elements of the same color
 - **Heuristic 1.2** – Calculate the Manhattan distance between each square and its closest neighbor of the same color
 - **Heuristic 2 – Number of squares:**
 - **Heuristic 2.1** – Calculate the number of groups of squares for each color that are not yet connected to each other
 - **Heuristic 2.2** – Calculate the number of squares that are not yet connected and adds up their Manhattan distance

Work already done

- **Programming language:** Python, using Pygame for the graphic interface
- **Development environment:** VSCode
- **Data structures:** Custom classes for different entities + more to come

- The game development is already in progress

- Demo link:

https://drive.google.com/file/d/1vmbYidO_sJHE37yMsZ9yVJeQC7nVtCPJ/view?usp=sharing



References

- Google Play - Cohesion Free
<https://play.google.com/store/apps/details?id=com.NeatWits.CohesionFree&hl=en&gl=US>
- Solver implementation for the 15 puzzle
<https://github.com/MichaelKim/15puzzle>
- Evaluating Search Algorithms for Solving n-Puzzle
<http://sumitg.com/assets/n-puzzle.pdf>
- Stack Overflow
<https://stackoverflow.com/questions>