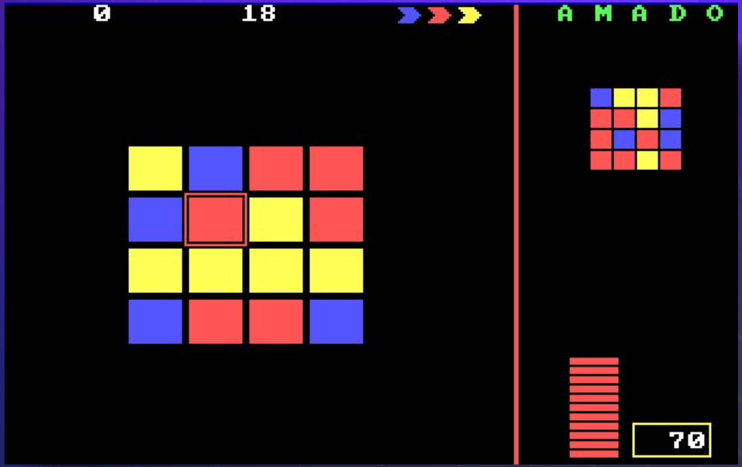
Work Specification - Topic 1



Group A1 33

Alberto Serra João Alves Eduardo Sousa

***Amado* Game**

Overview of *Amado* Game

Amado game is a challenging puzzle solitaire where a player manipulates a board of colored tiles with the goal of matching a predefined pattern and achieve the maximum number of points.

Relevance to Heuristic Search Methods

Heuristic search and optimization algorithms to automate the game. Solutions with the minimal number of moves and strategic hints, enhancing gameplay and decision-making for players.

Statistical Analysis for Optimization

Our project extends into statistical analysis, systematically evaluating which search strategies and heuristics yield the best outcomes for solving the Amado game's puzzles.

Related Work and References

01

Course Bibliography References

Sourced insights on heuristic implementations from "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, and "Reinforcement Learning: An Introduction" by Richard S. Sutton

02

Statistics on Search Strategies

Comparisons between uninformed search strategies like breadth-first and depth-first search with heuristic approaches from theoretical class and the internet

03

Technical Blogs and Tutorials

GeeksforGeeks, and Stack Overflow host articles and discussions on implementing and comparing search algorithms.

04

AI in Problem-Solving

Exploring AI technologies such as ChatGPT, Gemini, and Copilot in optimizing search strategies and heuristic approaches when coding.

05

Open Source Implementations

Exploration of existing open-source projects that implement puzzle games, serving as practical references for our development process (github)

Problem Formulation as a search problem

01

Names

Board configurations and cursor positioning to represent game states, aiming to align the current setup with a target state using strategic movements.

02

Effects

Movement operators (up, down, left, right) defined with uniform costs, pivotal in evaluating each action's effectiveness in progressing towards the goal state.

03

Preconditions

Evaluation functions for move assessment, focusing on reducing steps to the goal. Essential for the "Auto-Complete Game" feature, ensuring the shortest path to victory.

04

Costs

Integration of mutation and crossover functions with heuristics, for the "Hint System" and "Optimal Move Calculation" features, optimizing solution paths and strategic advice for minimal move completion.

05

States

In the beginning, the player's board and goal board are generated at random, and the player's cursor begins at the upper left square.
The blue, red, yellow, and empty squares on the board are represented by the values **B**, **R**, **Y**, and “ “ which correspond to a two-dimensional array.
The game state is updated based on the last play made by the player/machine.

06

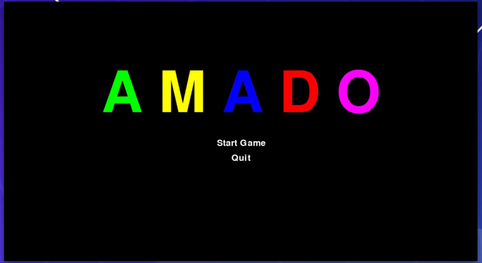
Objective Test

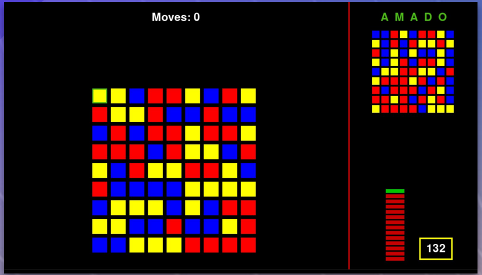
Verify that player’s board and the goal board are equal

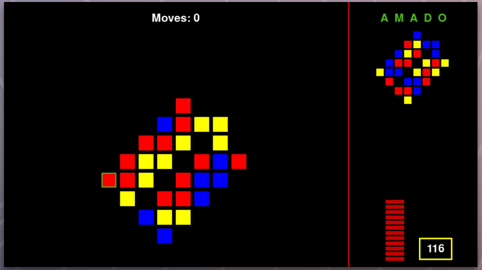
Work Done

We are using the Python programming language and the pygame library for game development. The project structure is organized as follows:

- run_gui.py - responsible for all the screen drawing and managing the game GUI (visual representation of the game elements and interactions with the player).
- board.py - all the implementations related to the game board (representation of the game board, the rules of the game, and any other board-related logic).
- game.py - where the game board is created and game states are controlled and updated (manages the game flow and any game-specific logic).
- solver.py - where all the search algorithms will be implemented. This is where the logic for finding the best move using search methods (ex: minimax algorithm and heuristics is housed). Responsible for providing hints and making optimal moves for the AI player.







Work to do

Least moves to win

The player will be given with the minimum number of moves required to win at the start of each game based on heuristic approaches. The best result from the heuristics will be shown.

Hint for best next play

It will be possible to request a hint at any position regarding the optimal play for that particular state. We'll employ search techniques that ensure the five next plays' depth.

Win game from any state

Because heuristic approaches run at a reasonable speed, it will be possible to ask the machine to solve any game, fully or partially. Both ways will show the actual gameplay.