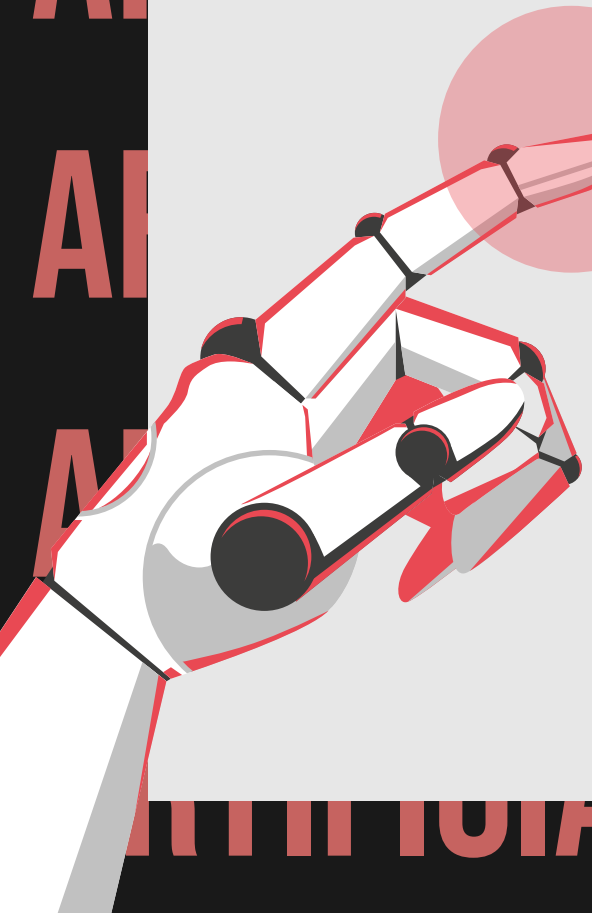


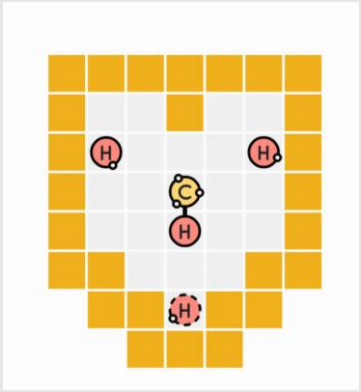
ARTIFICIAL INTELLIGENCE

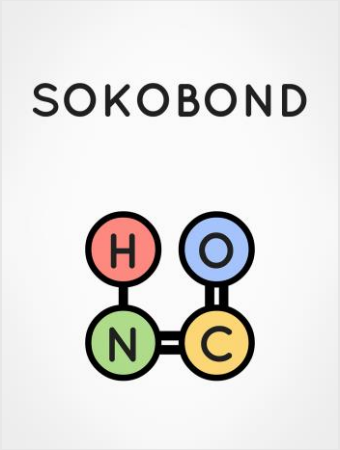
Sokobond - pygame

up202108689 – António Azevedo
up202108794 – José Martins
up202108776 – Tomás Martins

SOKOBOND

SOKOBOND OFFERS A UNIQUE PUZZLE EXPERIENCE THAT BLENDS LOGIC WITH CHEMISTRY. PLAYERS STRATEGICALLY MOVE ATOMS ON A GRID, AIMING TO FORM SPECIFIC MOLECULES BY BONDING WITH OTHER ATOMS LOCATED ON THE GRID





FOR THIS GAME, THE PLAYER IS EXPECTED TO BOND ALL THE ATOMS TOGETHER, FORMING THE DESIRED MOLECULE, WHILE ENSURING THAT NO ELECTRONS ARE LEFT UNCONNECTED

PROBLEM FORMULATION

STATE REPRESENTATION

CONFIGURATION OF ALL ATOMS ON THE GRID AS WELL AS THE AVAILABLE BONDING CONNECTIONS

INITIAL STATE

THE POSITIONS OF ATOMS ON THE GRID, THEIR BONDING CONNECTIONS, AND THE PLAYER-CONTROLLED ATOM.

OBJECTIVE TEST

EACH SOLUTION IS A SEQUENCE OF ACTIONS LEADING TO ONE MERGED MOLECULE WITH ALL BONDS UTILIZED

OPERATORS

- MOVE OPERATOR -

PRECONDITIONS

- CAN ONLY MOVE UP, DOWN, LEFT, OR RIGHT.
- MUST MOVE TO AN EMPTY CELL OR PUSH AN ATOM TO AN EMPTY CELL

EFECTS

- MOVES TO THE NEW CELL.
- FORMS A BOND IF ADJACENT TO ANOTHER ATOM AND BONDING LIMIT NOT EXCEEDED.
- PUSHES AN ATOM IF NO BONDING CONNECTION POSSIBLE.

COST

THE COST IS 1

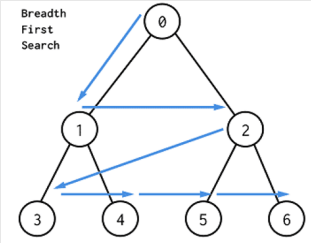
ARTIFICIAL INTELLIGENCE (AI)

UNINFORMED ALGORITHMS

01.

BREADTH-FIRST SEARCH (BFS)

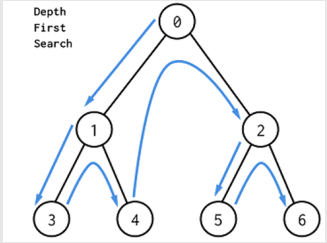
BFS explores the shortest path by examining all possible moves from the current state before moving to the next level of possibilities



02.

DEPTH-FIRST SEARCH (DFS)

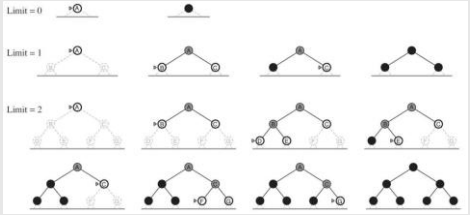
DFS explores as far as possible along each branch of the move tree before backtracking, prioritizing depth over breadth in its search for a solution.



03.

ITERATIVE DEEPENING SEARCH (IDS)

Iterative Deepening Search combines the benefits of both BFS and DFS by repeatedly performing DFS with increasing depth limits until the solution is found



HEURISTICS AND ALGORITHMS

ALGORITHMS

A* ALGORITHM

GREEDY SEARCH

HEURISTICS

MANHATTAN DISTANCE

PRIORITIZE FREE ELECTRONS

MINIMIZE FREE ELECTRONS

UNINFORMED ALGORITHMS - ANALYSIS

01.

BREADTH-FIRST SEARCH (BFS)

While it was anticipated that this algorithm would eventually reach the optimal solution, it was pleasantly surprising to observe that it achieved this outcome within a short timeframe without using too much memory resources.

02.

DEPTH-FIRST SEARCH (DFS)

In contrast to BFS, DFS can often find a solution, yet as the complexity of the level escalates, its memory demands can become quite high, guiding the search for an unfeasible solution. Despite not getting the optimal solution, DFS manages to produce results within a reasonable timeframe.

03.

ITERATIVE DEEPENING SEARCH (IDS)

While it was anticipated that this algorithm would surpass BFS in finding optimal solutions, it fails due to its poor search pace and inability to operate within memory constraints, often failing to find solutions within feasible limits.

INFORMED ALGORITHMS - ANALYSIS

01.

A* ALGORITHM

Regardless of the chosen heuristic, the A* algorithm consistently impresses by consistently discovering optimal solutions while excelling in both search time and efficient utilization of memory resources.

02.

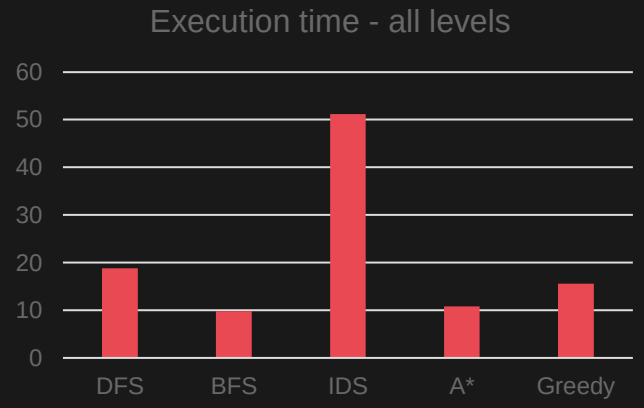
GREEDY SEARCH

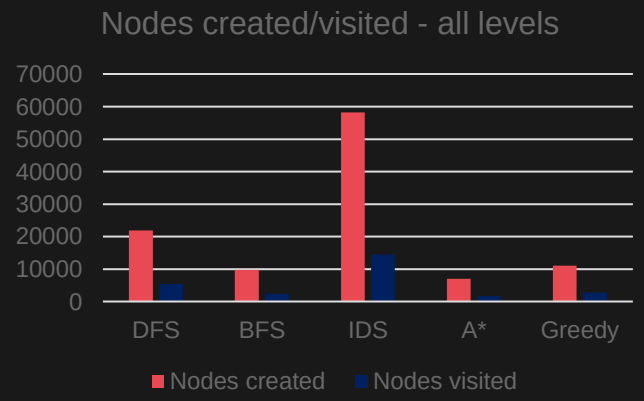
This algorithm was able to frequently find optimal or near-optimal solutions in a short time, regardless of the chosen heuristic. Unfortunately, challenges arise with highly complex levels, where the algorithm struggles to operate within memory limits, impeding its ability to find solutions.

ARTIFICIAL INTELLIGENCE (AI)

ALGORITHMS - ANALYSIS

- Both the Breadth-First Search (BFS) and A* algorithms excel in finding optimal solutions while maintaining impressive speed being the fastest and most capable of all algorithms
- The significant number of nodes stored in memory directly influences the time required for the Iterative Deepening Search (IDS) algorithm to reach a solution.





HEURISTICS - ANALYSIS

01.

MANHATTAN DISTANCE

This heuristic prioritizes states where the controlled atom/molecule is nearer to a potential bonding atom. The Manhattan distance heuristic exceed expectations, delivering excellent results even with the greedy algorithm, which can sometimes struggle to find feasible solutions.

02.

PRIORITIZE FREE ELECTRONS

This heuristic prioritizes bonding by connecting the player molecule to atoms with more free electrons. While not as promising as the Manhattan distance heuristic, it still achieved good solutions. However, when paired with the greedy algorithm, it often struggled to calculate feasible solutions.

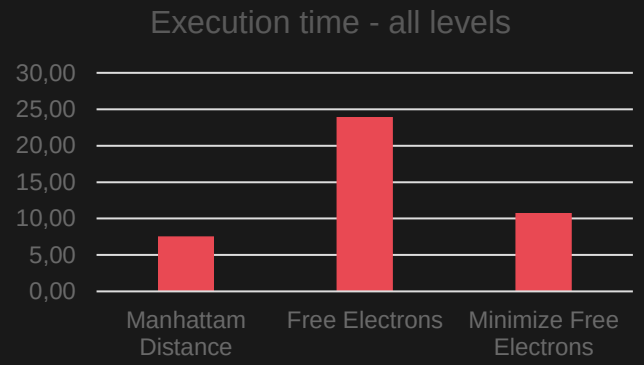
03.

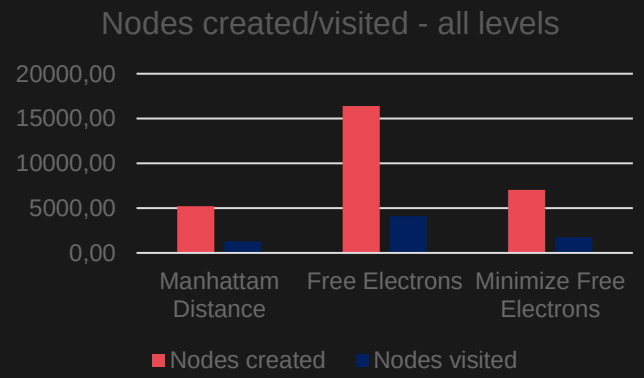
MINIMIZE FREE ELECTRONS

This heuristic evaluates the number of free electrons and favors states with fewer free electrons. While it operates within a reasonable search time frame, solutions generated may not be as optimal as with other heuristics. Additionally, this heuristic when paired with the greedy search can struggle to find feasible solutions.

HEURISTICS - ANALYSIS

- The Manhattan Distance heuristic consistently outperformed the others with both algorithms, delivering superior results in execution time, memory usage, and finding mostly optimal solutions;
- The Free Electrons heuristic, despite demanding a significant amount of memory resources and correlating with longer execution times compared to the third heuristic, yields substantially better solutions without compromising too much of the overall performance.





CONCLUSION AND SOURCES

This project provided us with a valuable opportunity to understand how search algorithms work in the context of single-player games. We gained a clear understanding of the importance of optimization, ensuring the usage of algorithms with a balance between solution quality, system stability, and time efficiency. Moreover, we understood the significance of developing effective heuristics to speed up the solution finding process. Overall, this project strengthened our comprehension of game development, algorithm optimization, and problem-solving techniques.

LINKS

STEAM PAGE: SOKOBOND ON STEAM;

ALTERNATIVE IMPLEMENTATION: JAVASCRIPT VERSION ON GITHUB;

AI UC BOOKS:

- STUART RUSSELL, PETER NORVIG: "ARTIFICIAL INTELLIGENCE" ISBN: 978-0-13-207148-2;
- RICHARD S. SUTTON: "REINFORCEMENT LEARNING" ISBN: 978-0-262-03924-6;
- STUART RUSSELL, PETER NORVIG: "ARTIFICIAL INTELLIGENCE: A MODERN APPROACH“;

PYGAME DOCUMENTATION: OFFICIAL PYGAME DOCUMENTATION