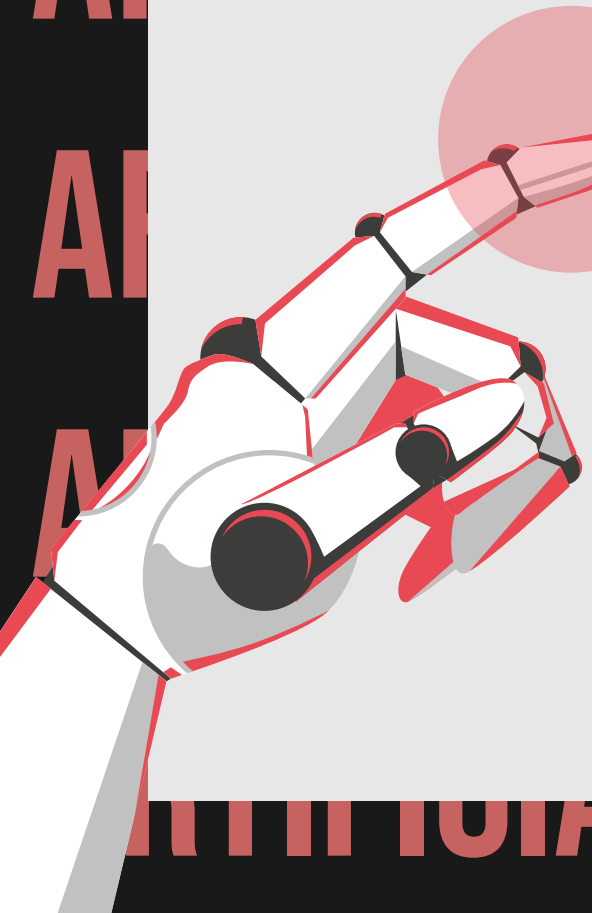


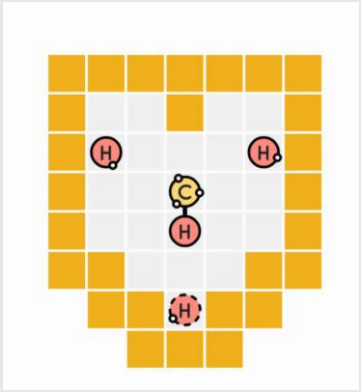
ARTIFICIAL INTELLIGENCE

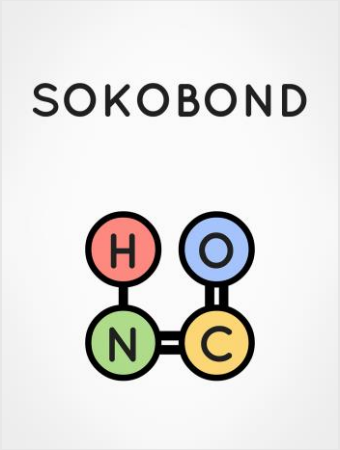
Sokobond - pygame

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

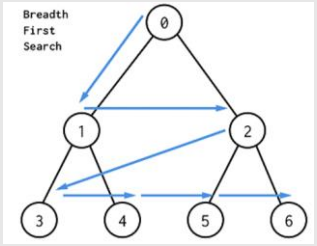
ARTIFICIAL INTELLIGENCE (AI)

UNINFORMED ALGORITHMS

01.

BREADTH-FIRST SEARCH (BFS)

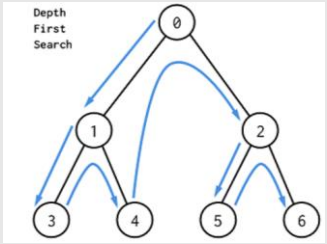
BFS explores the shortest path by examining all possible moves from the current state before moving to the next level of possibilities



02.

DEPTH-FIRST SEARCH (DFS)

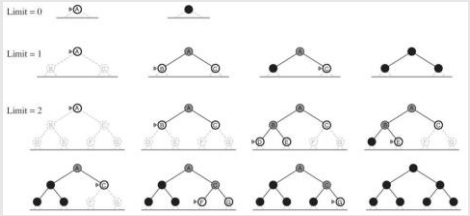
DFS explores as far as possible along each branch of the move tree before backtracking, prioritizing depth over breadth in its search for a solution.



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

A* ALGORITHM

- Manhattan Distance heuristic;

A* ALGORITHM W/PRUNNING

- Manhattan Distance heuristic;
- Prunes non-goal children;

GREEDY SEARCH

- Prioritize atoms with more free electrons;

A* ALGORITHM MINIMIZING LINKS

- Minimize remaining connections;

WORK DONE AND SOURCES

Engine: Pygame-powered (*sokobond.py*). Handles rendering, input, and state;
Loop: In *sokobond.py*. Controls FPS, updates, and rendering;

Data:

- ***Atom*** class (*entities.py*): Defines atoms and bonding.
- ***Game*** class (*game.py*): Manages game state and conditions.

Rendering: Using Pygame in *sokobond.py*, with atom sprites.
Input: Arrow keys defined in *sokobond.py*.

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