ATOMIX

Checkpoint 1

José Cunha - up201905451

Raquel Carneiro - up202005330

Tomás Pires - up202008319

SPECIFICATION OF THE WORK TO BE PERFORMED

Atomix takes place on a playfield consisting of a number of walls, with the atoms scattered throughout. The player is tasked with assembling a molecule with the atoms. The atoms must be arranged to exactly match the molecule displayed on the left side of the screen.

The player can choose an atom and move it in any of the four cardinal directions. A moved atom keeps sliding in one direction until it hits a wall or another atom. Solving the puzzles requires strategic planning in moving the atoms, and on later levels with little free space, even finding room for the completed molecule can be a problem.

Once the molecule is assembled, the player is given a score. The faster the puzzle was completed, the higher the score.

REFERENCES

In order to solve this state space search problem we can implement the heuristic algorithm A* and the limited memory algorithm IDA*. Their research led us to articles that explain and develop the algorithms:

- https://wayback.archiveit.org/all/20120712190655/http://www.user.tuberlin.de/hueffner/hueffner-studienarbeit-atomix.pdf
- https://www.geeksforgeeks.org/a-search-algorithm/
- Powerpoints presented in class

Related Topics:

- https://en.wikipedia.org/wiki/Atomix_ (video_game)

FORMULATION OF THE PROBLEM

State representation:

- 14 width x 13 height map, defined either by an empty space (0), a wall (-1), or an atom(1~n)
- Atom numbers are IDs that define their shape (element+connection directions); there may be repeated IDs.

Operators:

- All operators have a cost of 1.

Move atom up:

- Precondition: Space [xa,ya-1] = 0; xa,ya being the cooridnates of the chosen atom
- Effects: ya -= 1 until ya=0 or [xa,ya-1] != 0

Move atom down:

- Precondition: Space [xa,ya+1] = 0; xa,ya being the cooridnates of the chosen atom
- Effects: ya += 1 until ya=13 or [xa,ya+1] != 0

Move atom left:

- Precondition: Space [xa-1,ya] = 0; xa,ya being the cooridnates of the chosen atom
- Effects: xa -= 1 until xa=0 or [xa-1,ya] != 0

Move atom right:

- Precondition: Space [xa+1,ya] = 0; xa,ya being the cooridnates of the chosen atom
- Effects: xa += 1 until xa=14 or [xa+1,ya] != 0

Objective state:

- Defined by a sub-map describing a pattern of atoms that is smaller than the main map;
- Each space may either contain an atom (1~n) or nothing (0).
- All the atoms on the main map must match this pattern to reach the objective state.

IMPLEMENTATION WORK

Programming Language:

- Python

Development Environment:

- Visual Studio
- Anaconda

Algorithms to Implement:

- A*Algorithm
- IDA* Algorithm
- Greedy Algorithm

APPROACH

Evaluation Function:

The function receives a vector with the path from the initial state to the solution and returns its length minus one, which represents the number of moves made.

```
def eval_function(path):
    return len(path) - 1
```

Heuristics:

Given that the levels' solution can be achieved in more than one place, the solution's point of reference is given as a relative position from a given atom; as such, our heuristic of choice is given as the number of atoms not in this relative position.

IMPLEMENTED ALGORITHMS

- BFS
- Greedy Algorithm

EXPERIMENTAL RESULTS

The first level was concluded using the BFS algorithm, having a total of 365 moves to reach a solution and around 9 seconds to reach it.

Similarly, the first level was concluded using the Greedy Algorithm with a heuristic based on the number of atoms already in a relative correct position, taking a total of 124 moves and 0.148 seconds to reach the solution.

CONCLUSIONS

The biggest challenge of this project was trying to make the algorithms to run as smoothly as possible.

The rules of the game made the implementation of algorithms a difficult task, since every atom needs to be in a certain position to be accepted as a solution.

Then, the group considered that the elimination of steps towards the solution wouldn’t be beneficial because even when an atom finds its place, it might need to be moved so that another atom gets into place.

REFERENCES

- https://favtutor.com/blogs/breadth-first-search-python
- https://alan.draknek.org/games/atomix-levelpack/?levelpack=original