CSC442: Artificial Intelligence
Project 1: Search

September 9, 2021

Revisions

- Sep 9, 2021 - First Release

Overview

This project is about designing and implementing a program to solve a state-space search problem – specifically, to solve sliding block puzzles. This project has three components:

1. Design and development of your search program.
2. Experimentation and evaluation of your search program.
3. A project writeup.

You may complete this project alone or as a team of size two.

Project Requirements

You should carefully design your state-space representation. You will need to consider each of the following:

- State: how will you represent the puzzle board.
- Actions: the puzzle "blank" can be moved in a maximum of four directions (up, down, left, right); however, not all actions are always available (e.g., at the edges of the board.)
- Transition model: How will a move update your board?
- Initial state: the initial state will be given to you as console standard (plain text) input.
- Terminal states: the goal is to get all the non-blank tiles into numerical order, starting from top left. Although a real sliding block puzzle allows two final states, where the blank can be either top left or bottom right, for this project you should consider only one final state: where the blank is in the top left.

Finding Solutions

Your objective is actually to find the *shortest* solution. I.e., the solution to the puzzle with the fewest number of steps. Therefore, once you have developed a workable model of the puzzle, you must implement at least two search algorithms:

- Breadth First Search (BFS)
- AStar Search

If you choose to work as a duo, you must additionally implement a flourish, which goes beyond the basic requirements of the project. For this assignment, you may choose from one of three options:

- Create and use a pattern database heuristic.
- Implement Iterative Deepening AStar.
- Implement Simplified Memory-Bounded AStar.

To represent a solution, please output a series of moves with respect to the motion of the blank tile (e.g., a series of UP, LEFT, RIGHT, DOWN tokens in plain text – for more information, please see included examples.)

Evaluating Your Program

You should apply your program to each of the supplied test cases, working your way from the "easy" set to the "challenging" set, and identify which puzzles your program can effectively solve.

For this purpose, we'll call a puzzle "effectively solved" if your program can identify the shortest path in less than about a minute. Of course, you are free to let it run overnight on some of the tougher ones if you're curious, but this is not required.

Writeup

Lastly, you must produce a short PDF writeup with at least the following four sections:

1. Program design: Describe the overall architecture of your program, describe state/action representations, etc.) The purpose of this section is to help us understand how your program works so that we can grade it. In this section you should also describe the heuristics used in implementing AStar.

2. Collaboration: Describe each team member's contributions to the project.

3. Results: Describe how well your program works on the supplied test cases. Be sure to clearly specify results for BFS and AStar, as well as the effectiveness of your chosen flourish if you work as a team.

4. Discussion: Describe what you learned, was anything surprising, most interesting challenge, etc.

Your complete writeup should probably be in the range of 2-5 pages total. We will be grading on completeness, not length.

Submission Instructions

Upload a ZIP archive with everything (source code and writeup) to BlackBoard before the deadline.

Place all your source code into a directory called `src`. You must also include a plain-text `README` file in your src directory which describes how to build and run your program, and includes the names and UR NetID's for each contributing team member. Zip your `src` directory, `README`, and PDF writeup into a single file and upload to blackboard.

Submissions will be accepted until 11:59PM, Sunday, September 26th.