Assignment 2: Sample World Planner

Due: June 1, 2023 at 11:59PM

1 Assignment 2: Sample World Planner Specification

1.1 Overview

This assignment extends the solving approaches to the previous assignment. The problem description remains the same – you will implement two additional algorithms to solve the same Sample World problem.

Find the starter files and repository at `https://classroom.github.com/a/LdJdq-Eb`.

1.2 Running the Program

There are a few command line argument changes from the last assignment. Your program should accept up to two command-line arguments:

algorithm: one of **dfs** (depth-first search with cycle detection), **ucs** (uniform cost search), **ids** (iterative deepening depth-first search), **astar** (A*).

heuristic: if the algorithm specified is **astar**, include a second argument: one of three heuristic names: **h0** ($h(n) = 0$), **h1** (a better heuristic of your design), or **h2** (an even better heuristic of your design). All heuristics must be admissible.

Not every world can be solved in a reasonable amount of time with every algorithm.

1.3 Execution

You should write your code in one of two languages: Java or Python. You should name your source file that has the program's main method `SampleWorld`. One of the following options should invoke your program:

```
> python SampleWorld.py astar h1 < small1.txt
```

(Assuming the code has been compiled)

```
> java SampleWorld astar h1 < small1.txt
```

Also included is a utility program that, given a world and a solution, verifies that the solution reaches a goal state:

```
> java -jar sw-validator.jar -f /path/to/small1.txt
```

This utility takes a filename with the -f flag, and reads in your solution from standard in.

1.4 Design Ideas

When designing your A* heuristic functions, consider what costs are unavoidable from the current state.

2 Submission

Submit the code for your solution along with a writeup that answers the following questions

1. Describe the two heuristic functions you designed for A*, and describe why they are admissible.
2. Which of your implemented algorithms are admissible?
3. Describe choices you made in your code that you feel are important. Mention any specific aspects of your implementation that might be interesting as I evaluate your program.

4. What suggestions do you have for improving this assignment?

For submission, commit/push your updates to your Github repository. I automatically have access to the repository, so if you see your updated files on Github, I have access to them too.

3 Evaluation

- +15%: Your writeup is clear and descriptive.
- +25%: Depth-first search and uniform cost search are implemented efficiently and correctly.
- +25%: Iterative deepening search is implemented efficiently and correctly.
- +10%: A* works with only heuristic h0.
- +10%: A* works efficiently with one of your designed heuristic functions.
- +15%: A* efficiently produces optimal plans with both heuristic functions.