Foundations of AI - Assignment 3 - The Elevator Problem

Author: Dominic Soares

Last Changed: 1.11.18

Description:

This program simulates people moving throughout a hotel building with multiple floors and elevators. The people want to move between floors by using the elevators. Using the information about each person and where they want to go we can simulate the elevators moving and search through the simulated states of the hotel to find the movements that bring the people to their destination floor.

Description of Algorithm:

The hotel object is comprised of the floor, elevator, and people objects. Each floor has a certain number of elevators and people on it, and each elevator has a certain amount of people on it. After each person is added to the system, they are inserted into a floor in the main state of the building and the search algorithm begins, trying to find the set of moves that transfer the person to their destination. When the moveset is found, the system follows that movement until another person is added to the system. This process is repeated until all people are at their destinations and no more people are added to the system.

Usage:

python2 hotel.py

Modifiable Pieces:

[line number, type, description]

16. Secure Floors – The numbers represent the secure floors of the building

22-25. Elevators – The values passed in represent the capacity, floors it services, name, starting floor

383. Main object instantiation – the value passed in is the total number of floors

                - must also change second number in line 464 and 475

396. First person – The values for the first person added to the system represent their starting floor, type of person (e – employee, v – vip, g – guest), name (don’t change), goal floor

434. Total time – The number represents how many iterations the program does before stopping

460. Create person time – The number represents the time at which no more people will be added to the system

461. % change add person – The second number is the denominator for the chance a person is added to the system each iteration. E.g. 60 = 1/60 chance

467. Type of person – The odds for the people are modifiable based on the following if else statement

Useful Modifiable Print Statements

[line number, type, description]

406. Print state - prints the state of system directly after adding a person

419. Print solution length - prints the length of the current solution moveset

421,422. Print states - prints the beginning and end state of a solution

429. Print state - prints the next state after a move forward

438. Print time - prints the iteration number

547. Print heuristic value - prints the heuristic value for the printed state

548. Print heuristic flag - prints the heuristic flag for the printed state