

Introduction to Artificial Intelligence

Assignment 1: Book Finding

Deadline

Week 9 (16.03.19 23:59)

Submission

- Only one Java file with the code has to be uploaded to Moodle. Input, output and other files are NOT allowed. Name of the file should be like this *NameSurname.java* (For example, *IvanIvanov.java*). No other symbols allowed
- Report describing each algorithm flow in plain English (not more than 2, 3 paragraphs), statistical comparison among algorithms and PEAS description with respect to the Actor agent. Name of the file should be like this *NameSurname.pdf* (For example, *IvanIvanov.pdf*). No other symbols allowed
- Also, report should include Graphical representation of maps that were impossible to solve. You can generate the maps directly through the code or hand draw them for the report after testing them on both algorithms
- Report may include any interesting outcome/map worth highlighting

Programming Language

Java (JDK 1.8)

Requests

- The program must work, the code should be readable, well-structured and should contain JavaDoc
- It has to be only one **.java* file and report **.pdf* file
- It is allowed to use only standard Java libraries
- NO extension of a deadline. Works sent after the deadline will NOT be evaluated
- Assignment is individual
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies

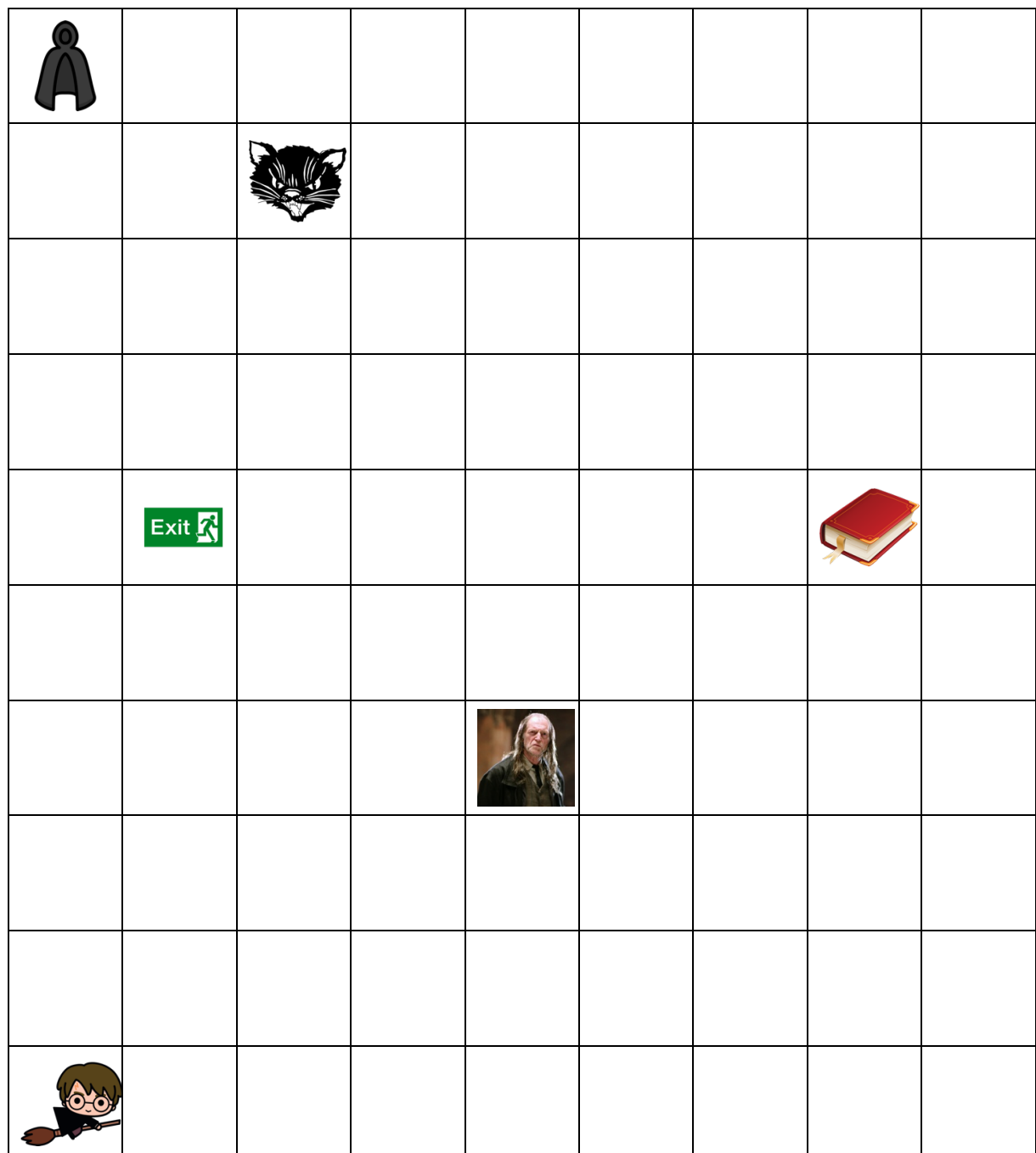
Grading Criteria

- 60% for the code correctness
- 10% for readability of code and JavaDoc
- 30% for the well-structured and informative report

Task

According to the bestseller of Joanne Rowling called Harry Potter and Philosopher's Stone, the main hero was looking for a book about Nicholas Flamel, the only known creator of a philosopher's stone. However, the book was located in the restricted section of a library. Therefore, Harry has decided to use his invisibility cloak to hide from unexpected guests and go to the library during the night. However, he has lost his invisibility cloak somewhere in the library while looking for a book. At that time, Argus Filch and his cat Mrs Norris began inspecting the library. Harry has to find the book and leave the library without being caught.

Your environment is a 9*9 square lattice.

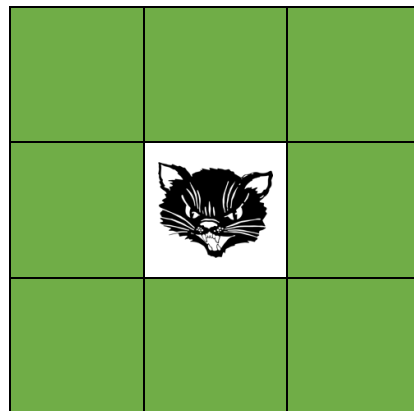


Actor

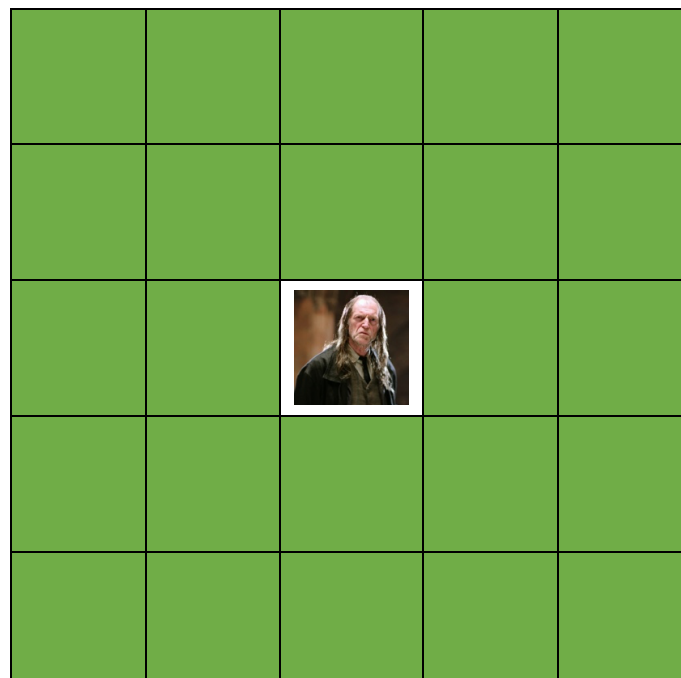
You start from bottom left. Your goal is to reach door in as minimum number of steps as possible. **Your ability to perceive inspectors is defined in the “*variants section*” below.** Your algorithms will work on both variants. The actor can move one step per turn and can move horizontally, vertically and diagonally.

Inspectors

1. Mrs Norris: Cat’s perception is only in consecutive cells (Moore neighborhood), shown in figure below.



2. Argus Filch: His perception is Moore neighborhood of range 2 as he has a candle, shown in figure below



Both of them are generated randomly on the map. You do not want to face inspectors as it ends the game. You can decrease the perception zone of inspectors to 1 cell (their location) by finding the invisibility cloak. Technically you will be invisible to them until you collide with them

Exit

The exit door from the library is randomly generated on the map except inside the inspectors' cells and except the cell where the book is allocated. You know the location of the exit door.

Book

The book is generated randomly and is not in the inspectors' zone. You do not know the location of the book. You can perceive the book only when you are inside the book cell.

Invisibility Cloak

The invisibility cloak is generated randomly and is not in the inspectors' zone. You do not know its location. You can perceive the invisibility cloak only when you are inside its cell. If you found it, inspectors' perception decreases till the cell, where they are located

Algorithms

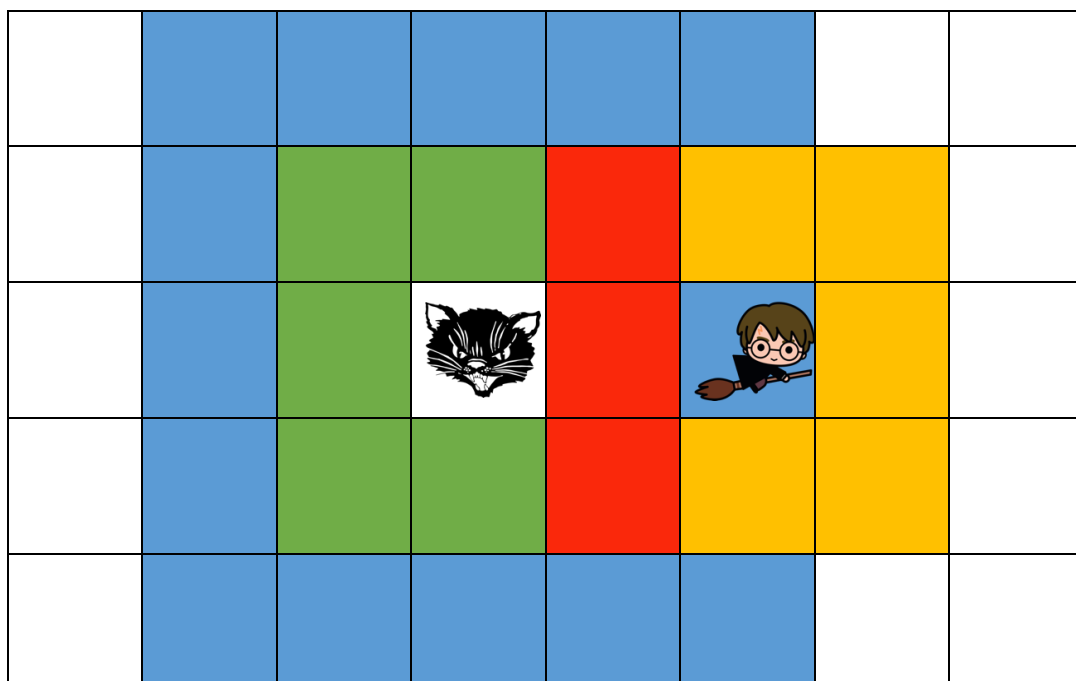
- A backtracking search
- Another search method of your choice (e.g., A*, Simulated Annealing, etc.)

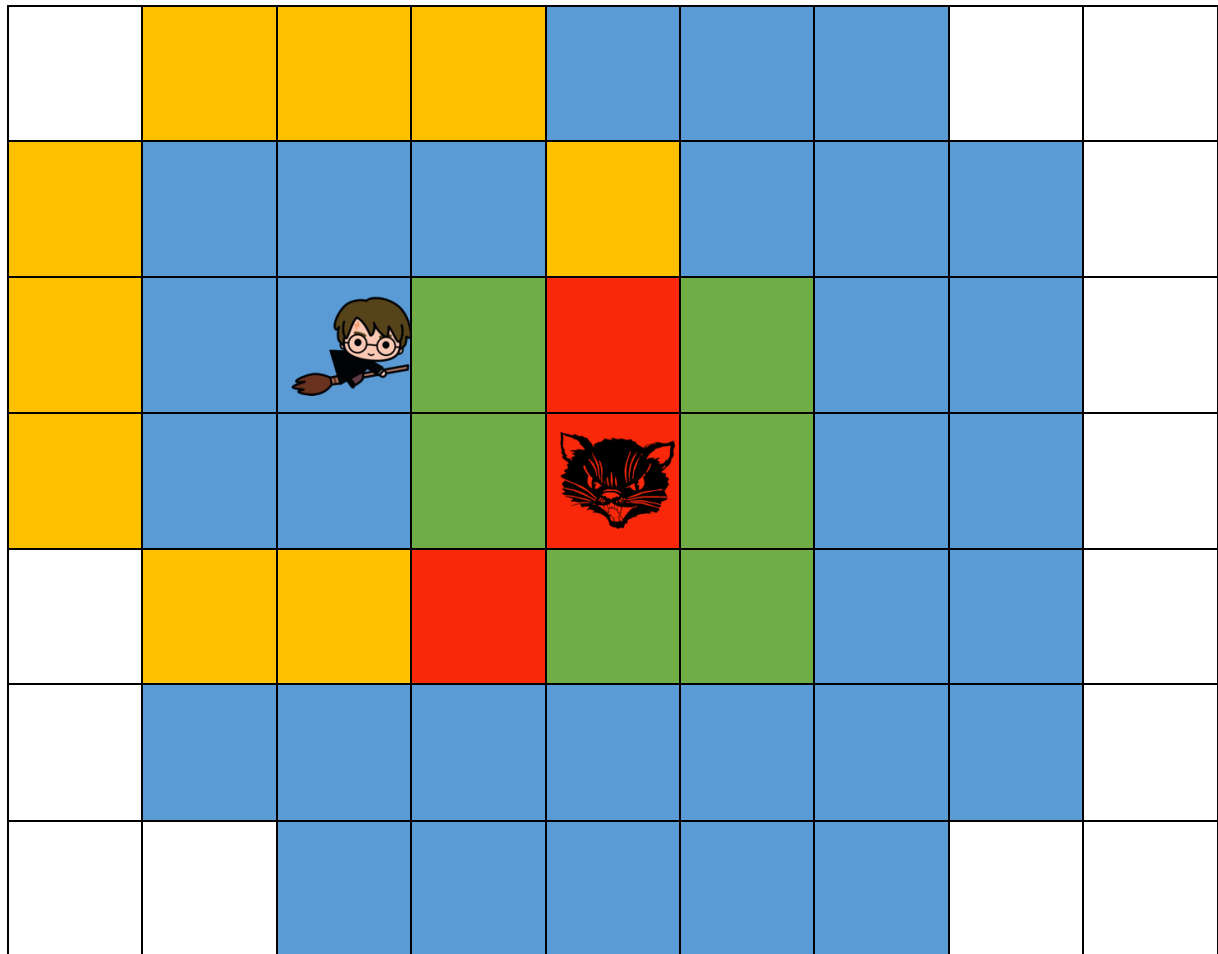
You are allowed to use modified versions of algorithms, do not forget to describe modifications in the report, if there will be any

Variants

The algorithms consider two scenarios:

1. You can perceive the inspector zone's adjacent cells if you are standing next to their perception zone (in blue cells), shown below. In the first figure, you are able to perceive 3 red cells, in the second only 1. Orange cells are the actor's perception of the inspector's zone





Consider that wearing an invisibility cloak does not change your perception distance

Input

The algorithm's input is a 9*9 square lattice. The map has a single actor, Argus Filch, Mrs Norris, the book, the invisibility cloak and an exit door. The input file would be a sequence of positions for each agent in the abovementioned order followed by the actor's perception scenario. Possible positions are including [0,0] [0,1] [0,2]....[8,6] [8,7] [8,8]. Possible scenarios are 1 and 2.

It should be possible:

1. to generate the map and manually insert perception scenario
2. to insert the positions of agents and perception scenario manually.

While generation guarantees valid inputs, manual typing may lead to errors. Any incorrect inputs violating conditions of the assignment should lead to the error message and request for valid data. Input should be written in 2 strings. For example,

[0,0] [0,5] [2,7] [7,5] [0,8] [1,5]

1

describes the positions of all actors in the first figure:

- [0,0] – Harry
- [0,5] – Argus Filch
- [2,7] – Mrs Norris
- [7,5] – Book
- [0,8] – Invisibility Cloak
- [1,5] – Exit

and defines that the first scenario of actor's perception is chosen.

Output

The output comprises of

1. Name of the algorithm
2. Outcome - Win or Lose
3. The number of steps algorithm took to reach exit door
4. The path on the map. Path can be displayed as, for example, [0,0] [1,1] [1,2]...
5. It would be better if the path is highlighted on the map. You can use console for it
6. Time taken by the algorithm to reach the exit door

There should be 2 output sequences for each input – one for each algorithm.

Statistical Analysis

Comparison of algorithms through statistical arguments based upon test maps generated. Statistical analysis is required for both variants (described above). As an example , for each test map, comparison would be:

- Backtracking (variant 1) compared to 2nd algorithm (variant 1)
- Backtracking (variant 2) compared to 2nd algorithm (variant 2)
- Backtracking (variant 1) compared to Backtracking (variant 2)
- 2nd algorithm (variant 1) compared to 2nd algorithm (variant 2)