CSCI-580 Project Wumpus

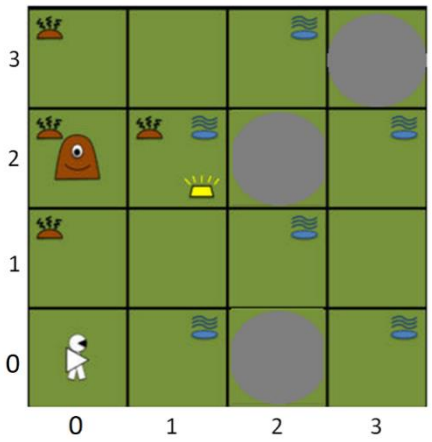
Objective: In this project you will implement a logical agent to solve a Wumpus world puzzle.

Input:

An integer N on the first line, and then N x N matrix of observations, each is a string, starting with the observations at (0, 0) cell and ending with (N-1, N-1) cell. If an input observation has the character 'S', then the corresponding cell has *Stench* in it; if it has 'B', then the cell has *Breese*; if 'G', then the cell has a shining *Gold* in it. The observations that have no perceptual input are given by a string "N".

Input Example:

For the Wumpus World given on Figure below, you will have the following input:



4

N B N B (for the row 0)

S N B N (for the row 1)

S SBG N B (for the row 2)

S N B N (for the row 3)

Note: you are only given perceptual observations, there is no information about Wumpus or Pits in the input.

Output:

If it is possible to safely retrieve the gold starting at the cell (0, 0), then your program will output the path consisting of the pairs of cells' coordinates.

Format is:

<(row,col)><space>…<(row,col)><space><endl>

Output Example (for the Figure above):

(0,0) (0,1) (1,1) (2,1)

If it is not possible to find gold, then your program will print out the following message:

```
cout << "It is not possible to find Gold." << endl;
```

The rules of the Wumpus World:

1. If a cell has a Pit, then adjacent cells to the North, South, East and West have Breese.
2. If a cell has a Wumpus, then adjacent cells to the North, South, East and West have Stench.
3. Each cell can either have a Pit or a Wumpus, but not both.
4. There is exactly one Wumpus in a given Wumpus World.

Specification Requirements:

1. Use dfsVisit on (0, 0) cell to find a path.
2. Your algorithm must make inference only on the observed so far perceptions. In other words, keep input data separately from the Knowledge Base data obtained by making observations of visited cells.
3. If a current cell has row = i and column = j, then process the children in this order: (i-1, j), (i+1, j), (i, j-1) and (i, j+1), i.e. in the order of: North, South, West, and East directions.
4. Once all children have been processed, and gold has not been found, then make inference on pairs of children of the current cell, if certain pairs of children both have Stench, then you may infer Wumpus and need to re-run dfsVisit on the current node. To be able to do this, write another dfsVisitReset function on the current cell: this function will reset all nodes visited after current node to White color. Simply visit only those safe children whose color is Black and whose parent is current node. First, call dfsVisitReset on the current cell (it will reset all nodes visited after the current cell to White color), and then call dfsVisit on the current node again. **The pairs of children that may infer Wumpus are:** (child N and E), (child N and W), (child S and E) and (child S and W), where N, S, E, W are directions North, South, East and West. If both children in a given pair have Stench, then we may conclude that there is the Wumpus at the diagonal cells in the direction NE, NW, SE, and SW respectively.
5. Write a class Wumpus, but put it into *main.cpp.* Inside *main()* function, you need to instantiate a Wumpus object and call member functions of this class to achieve the goal.
6. ***Do not place the entire code into main() function.***

Submission:

Submit *main.cpp* to turnin.

Overview of Inference steps:

1. If observation is *No Stench*, $\sim S_{i,j}$, then set all neighboring cells of Wumpus matrix to *False*
2. If observation is *No Breeze*, $\sim P_{1,1}$, then set all neighboring cells of Pit matrix to *False*
3. If observation is *Stench*, $S_{i,j}$, then **check** all neighboring cells of Wumpus matrix:
    1. If only one neighboring cell is not set, and the rest are set to *False*, then Wumpus must be in the cell whose value is not assigned
    2. If Wumpus is found in one cell, then the rest of the cells have no Wumpus
    3. If Wumpus is found in a cell, then set Pit of that cell to *False*
4. If observation is *Breeze*, $B_{i,j}$, then **check** all neighboring cells of Pit matrix:
    1. If only one neighboring cell is not set, and the rest are set to *False*, then Pit must be in the cell whose value is not assigned
    2. If Pit is found in a cell, then set Wumpus of that cell to *False*
5. If a pair of children in direction (N, E) from current cell have stench, then the cell in diagonal direction NE has Wumpus (there are three more possible pairs of children with stench that can infer Wumpus)

Wumpus Worlds used in test files: