

Project #2

Wumpus World Solving

Due 10/5/98

Background

Agents which solve problems such as the Wumpus World can be classified into three types: reflex agents, model-based agents, and goal-oriented agents. Reflex agents respond to the percepts of the current position, and behave accordingly. Such an agent would likely never solve the Wumpus World problem. Model-based agents model the world internally, and try to determine the best strategy based on that model. Goal-based agents have a particular goal to achieve, and attempt to take actions which further that goal by some measure. Goal-based agents are somewhat difficult to implement.

Instead we will implement something of a model-based agent, which attempts to construct an internal model of the world, and then act accordingly. This model will be a simple representation of the Wumpus World, similar to the internal representation of the system. Note that these two representations should be distinct logically, even if the details of the code differ. That is to say that the agent should only act on information it can legitimately have.

Your Task

You have three particular tasks for this project:

Step 1: Place your code from project 1 in a package. Your new code will need to reference that code, but should remain distinct. You will not be graded on your old code again, except where it dramatically impacts your work for this project. Focus your efforts on this project.

Step 2: Create a second package containing a function to randomly generate valid Wumpus Worlds. A valid world will not have a pit or Wumpus on the start square, nor will the Gold be in a pit square. Any other combination is valid.

Step 3: Write functions that solve the Wumpus World problem.

Step 4: Write a function which generates a random world and solves it a specified number of times, n (make n a parameter), and then scores the agent accordingly. The agent should lose 500 points for dying, 5 points for missing the Wumpus with the arrow (should this occur), 2 points for moving, and 1 point for turning. The agent should gain 500 points for exiting with the gold, and 100 points for killing the Wumpus.

Hints

Solving the Wumpus World should follow the following algorithm, in descending order of preference (i.e. picking up the gold should always be done if possible).

- ❖ If 'Glitter is present in the current square, grab the gold.
- ❖ If you have the gold and are in the entrance, climb out
- ❖ If you have the gold, move to the entrance.
- ❖ If the Wumpus position is known, move to kill it if possible.
- ❖ If there is a safe unexplored square, move to and explore the closest such square.
- ❖ If there is no safe unexplored square and the Wumpus is dead, move to the exit, and climb out.
- ❖ If there is no safe unexplored square, explore the closest unsafe square (You may want to refine this further, to achieve a better performing agent -- some squares are more unsafe than others)

You may need functions to do the following, among others:

- Return a list of safe unexplored squares, sorted by distance
- Determine the shortest safe path between two positions
- Return closest unsafe unexplored
- Determine the Wumpus position, if possible, and return a list of squares from which the Wumpus may be killed, sorted by distance
- Execute an action, updating the score

Grading

Grading will be similar to the previous project, although more emphasis will be placed on style, that is to say that style points will be more difficult to come by.