

Washington State University
School of Electrical Engineering and Computer Science
Fall 2020

CptS 440/540 Artificial Intelligence

Homework 5

Due: October 8, 2020 (11:59pm pacific time)

General Instructions: Submit your files as an attachment under Content → Homework 5 for the course CptS 440 Pullman (all sections of CptS 440 and 540 are merged under the CptS 440 Pullman section) on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but we will only grade the most recent entry submitted before the deadline.

For this homework you will implement a hybrid problem-solving logic-based agent to play the Wumpus World game. The file “wumpus-search.zip” that accompanies this homework provides an A* search using the city-block heuristic. The search algorithm is implemented in the files Search.h, Search.cc and Search.py. The zip file also includes modified Agent.h, Agent.cc and Agent.py files that demonstrate the use of the search capability. Copy all these files, including the updated Makefile, to a copy of the simulator code and recompile to use the search capability. You will mainly need to use the following three methods of the SearchEngine class:

- `AddSafeLocation(x, y)` – Adds a location to the search engine’s list of safe locations. The search engine considers only safe locations in its solution.
- `RemoveSafeLocation(x, y)` – Removes a safe location from the search engine’s list of safe locations (e.g., in case a previously-thought safe location turns out to be unsafe).
- `FindPath(startLocation, startOrientation, goalLocation, goalOrientation)` – Returns a list of actions to get from the start state to the goal state. If no safe path is possible, then returns an empty list.

Your agent should do the following.

1. In addition to the agent’s state information, you should also keep track of **stench locations**, the **Wumpus location**, the **gold location**, any **pit locations**, **visited locations**, **safe locations**, and the **world size**, as this information becomes known. You will be playing each world for multiple tries, so keeping track of what you learned from previous tries is useful (and required).
2. You should update location information as you visit new locations:
 - a. If you perceive **a stench**, then add current location to stench locations.
 - b. If you perceive **a glitter**, then set gold location to current location.
 - c. Update the **world size to max(X,Y)** of your current location.
 - d. Add current location to **safe locations**. If **no breeze or stench**, then add adjacent **locations to safe locations**.
 - e. Add current location to **visited locations**.

- f. Implement the logical rule: If there are two stench locations in a diagonal arrangement, and one of their common adjacent locations is safe, then the other common adjacent location contains the Wumpus. For example, if there is a stench in (2,2) and (3,3), and (3,2) is safe, then the Wumpus is in (2,3). Your agent should never die from the Wumpus, nor need to shoot the Wumpus (the gold will not be co-located with the Wumpus)
 - g. If your agent dies, then it should only be from doing a GoForward into a pit, so update pit locations accordingly.
3. If you perceive a glitter, then **Grab**.
4. If you have the gold and are in the (1,1) location, then **Climb**.
5. If you know the **gold's location**, and you don't have the gold, then use the **search engine to find a sequence of actions** to get you there.
6. If you have the gold, but aren't in the (1,1) location, then use the search engine to find a sequence of actions to get you to (1,1).
7. Otherwise, **determine a safe unvisited location** and use **the search engine to find a sequence of actions to get there**.

Your agent should be implemented entirely in the Agent.h and Agent.cc files (for C++ implementations) or the Agent.py file (for Python implementations). You may also include an optional readme.txt file with any extra instructions for compiling and running your agent. Your agent should not require any user input. Your agent will be tested by copying only your Agent.h and Agent.cc files, or Agent.py file, into a fresh copy of the simulator code (that includes the new Makefile, Search.h, Search.cc and Search.py), and compiling and running it on several test worlds. Your grade will be based on satisfying the above requirements, good programming style, and your agent's average scores on the test worlds.

The test worlds will adhere to the following constraints:

- The world size can vary from 3x3 to 9x9.
- The gold will never be co-located with the Wumpus or a pit.
- There will always be a safe path to the gold.