Soren DeOrlow
IDSN 599, Fall 2021
deorlow@usc.edu
Homework 2

Strategies for developing an agent within the wumpus world environment.

	0	1	2	3	4	5	6	7	8	9
0	0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4	4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Tracking percepts

1. What data structure will you create to keep track of the percepts for each room that you have visited?

Assigning risk to grid lines

One approach would be to create a library and assign a risk level to each grid line between rooms, which would look like this (0,0:1 risk=x). Given the coordinates in wumpus world code, this approach is not possible.

Assigning risk to rooms

Another approach would be to track the rooms that have been visited and not visited, assigning risk levels for each room that has not been visited. For example if you are in the starting position and move forward one room and receive a "breeze" percept, then you would increase the risk factor of rooms 0,2 and 1,1 by a factor of 1, because at this point it would be impossible to know if one or both rooms had pits. The next step would be to travel to 1,0 to see if a breeze percept were present. If a breeze was not detected, 1,1 would receive a lowered risk factor of 1 and 2,0 would receive a risk factor of 1, while raising 2,0 to 3.

What an agent would record while travelling through the grid

The first move indicates a breeze percept at 0,1 rising the risk factor for 0,3 and 1,1.

0	0,B	2	1
1	2	1	1
1	1	1	1

The agent then moves back and down to confirm that 1,1 is not indicating a breeze. 0,3 is raised to 3.

0	0,B	3	1
0	1	1	1
1	1	1	1

A dictionary would record the following:

['0,0 ': '0', '0,1': '0' (breeze), '0,2': '3', '1,0': '0', '1,1': '1', '2,0': '1']

Assigning risk levels

Visited room: 0

Unvisited room: 1

Room adjacent to percept: 2

Room surrounded by 2 percepts: 3

Room surrounded by 3 percepts: 4

	Breeze	
Breeze	PIT	Breeze
	Breeze	

Grid location

A factor affecting the algorithm is grid location, it dictates how to allocate risk across adjacent rooms.

The wumpus world environment presents a grid of 10 x 10 rooms indicated in the following coordinates.

0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

The risk landscape changes depending on where the agent is in the grid.

Perimeter area:
Two risk location possibilities

A dictionary of risk factors might look like this:

Position: 1,0 [‘0,0 ’: ‘0’, ‘1,0’: ‘0’, ‘1,1’: ‘1’, ‘2,0’: 1]

0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Center area:
Three risk location possibilities

A dictionary of risk factors might look like this:

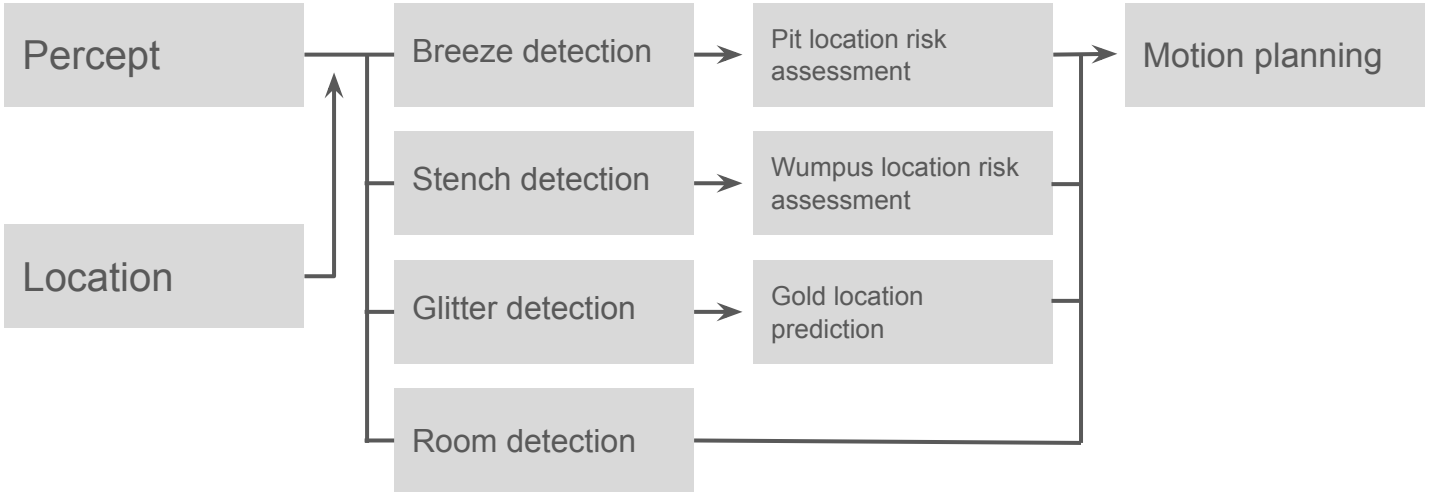
Position: 1,1 [‘0,0 ’: ‘0’, ‘1,0’: ‘0’, ‘1,1’: ‘0’, ‘2,1’: ‘1’, ‘0,1’: ‘1’, ‘1,2’: ‘1’]

0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Goal based decision-making

2. What algorithm will you use to iterate over the data structure from question 1 to assist you in making decisions as to where to move to next?

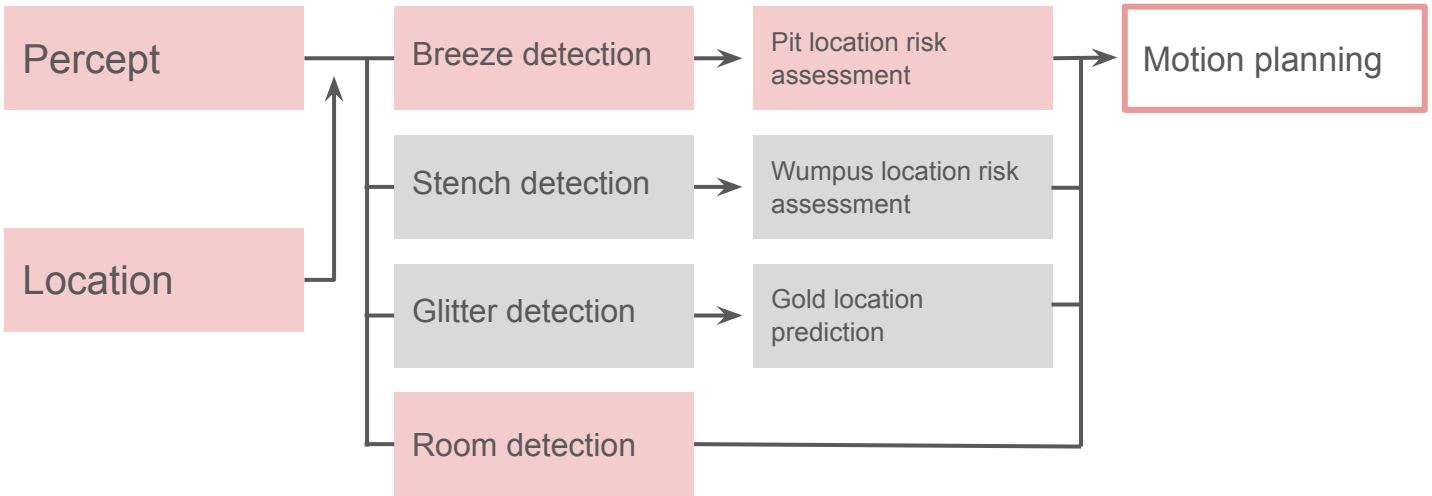
As the player travels around the board risk percepts become more clearly defined. By assigning various degrees of risk to each unknown room, it creates criteria for navigating uncertainty. Priority is given to low risk rooms versus high risk rooms. The player will pursue low risk rooms and gather information about the degrees of risk that other rooms possess. At certain times, a player will have to have to enter a room with a level 2 risk out of necessity.



Motion planning algorithm

3. What algorithm will you use to make decisions as to what to do next?

The player would proceed to move forward until it encounters a percept. If the percept is a breeze or stench, the player should turn around (TR, TR), move back in the previous direction one room and proceed to an unknown room with a risk level of 1.



Retracing steps

4. What data structure will you use so that once you get the gold you can retrace your steps to get back to room 0,0 and climb (CL) out of the cave?

This will require you to reverse sort the library to identify the path in reverse. It might be valuable to somehow create a global compass. The challenging aspect of going in reverse is player rotation controls. A formula will need to be created to convert the TR, TL controls to north, east, south, west in a way that would be reversible. This route would follow every step the agent took and may not be the most efficient route.

Retracing steps to the gold and back

The agent would follow this path to the gold:

[“FW”, ‘FW’, ‘TR’, ‘FW’, ‘FW’, ‘FW’, ‘FW’, ‘FW’, ’TL’, ‘FW’, ‘FW’ (breeze), ‘TR’, ‘TR’, ‘FW’, ‘TL’, ‘FW’, ‘FW’, ’FW’ ‘TL’, ‘FW’, ‘FW’, ‘GR’]

The agent would follow this path back:

[TR, TR, FW, FW, FW, TR, FW, FW, FW, TR, FW, (breeze), TR, TR, FW, FW, TR, FW, FW, FW, FW, FW, FW, TL, FW, FW, CL]

0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

ML motion planning

5. What algorithm will you use to retrace your steps using the data structure in question 3 to exit the cave once you have the gold?

With the gold in hand, the approach can change and the player will seek only rooms that are at risk level 0 on the path back to room 0,0. There would be no benefit to visiting 5,4 and it would be eliminated from the path. Given that 8,2-5,2 did not indicate a precept, it would be more efficient to take this path and limit turning. The agent could take a goal-based approach to find a more efficient path back out of the cave.

Retracing steps to the gold and back

The agent would follow this path to the gold:

[“FW”, ‘FW’, ‘TR’, ‘FW’, ‘FW’, ‘FW’, ‘FW’, ‘FW’, ’TL’, ‘FW’, ‘FW’ (breeze), ‘TR’, ‘TR’, ‘FW’, ‘TL’, ‘FW’, ‘FW’, ’FW’ ‘TL’, ‘FW’, ‘FW’, ‘GR’]

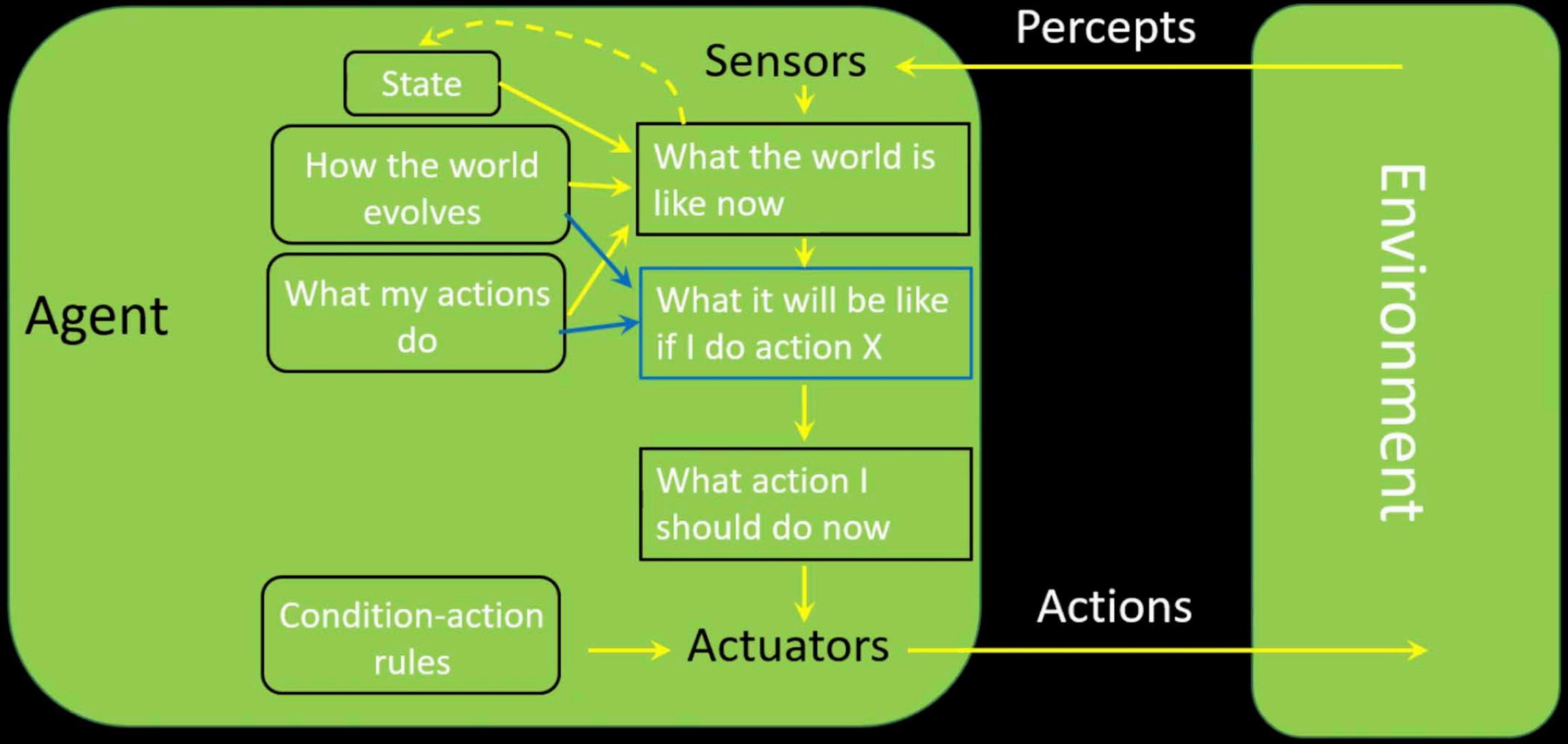
The agent would follow this path back:

[TR, TR, FW, FW, FW, TR, FW, FW, FW, TL, FW, FW, TR, FW, FW, FW, FW, FW, FW, TL, FW, FW, CL]

0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

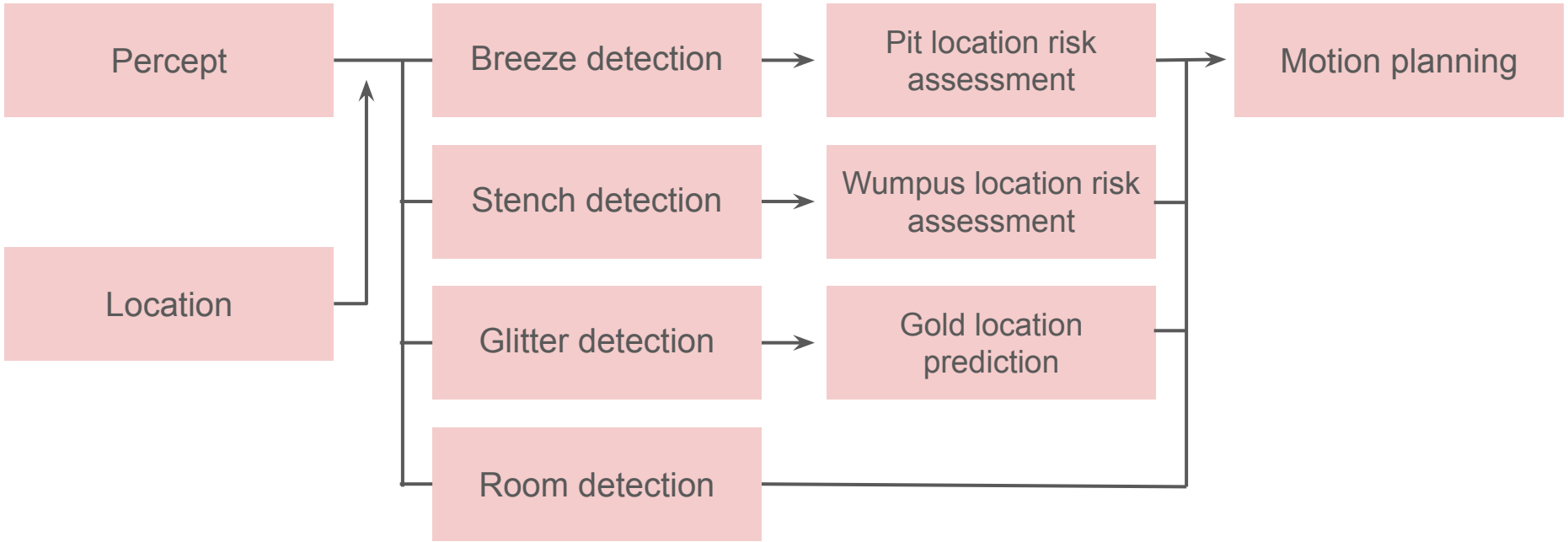
Appendix

Goal-Based Agents



Decision making

Steps for guiding agent within wumpus world.



Wumpus world grid

	0	1	2	3	4	5	6	7	8	9
0	0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4	3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Tracking percepts

Perimeter area: 2x risk location possibilities

A dictionary might look like the following:

Position: 1,0 [‘0,0 ’: ‘0’, ‘1,0’: ‘0’, ‘1,1’: ‘1’, ‘2,0’: 1]

	0	1	2	3	4	5	6	7	8	9
0	0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4	3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Center area: 3x risk location possibilities

A dictionary might look like the following:

Position: 1,1 [‘0,0 ’: ‘0’, ‘1,0’: ‘0’, ‘1,1’: ‘0’, ‘2,1’: ‘1’, ‘0,1’: ‘1’, ‘1,2’: ‘1’]

	0	1	2	3	4	5	6	7	8	9
0	0,0	0,1	0,2	0,3	0.4	0.5	0,6	0,7	0,8	0,9
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4	3,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9