CS 511 Project 2:

Model Based Agent for Wumpus World

By Wumpus Slasher

Classes:

Cell: The building block of the model. It records states of a traveled wumpus world coordinate. Each Cell corresponds to a unique coordinate of the wumpus world.
Detailed explanations are given as comments in Cell.java.

Model: This class is the core of the model based agent. All calculations are done in the model. It translates percepts into states and records it for later use by modifying the flags of cells and some internal variables. At each step the model is updated with given percepts, then the agent function determines the best next move based on the model.
Detailed explanations are given as comments in Model.java.

Agent Function: This class takes precepts as input and updates the model based on percepts at each step. The following are some rules used in agent function.

1. If glitter, the agent will grab the gold which will earn 1000 credits.
2. If bumped, it will turn left or right based on where the next safe cell and nearby wall are located.
3. If the agent feels the breeze at the very beginning, it will stop moving (return just Action.NO_OP). This gives about 3 points of improvement.
4. If the agent feels the stench at the very beginning, it will shoot an arrow toward the east which provides a 50/50 chance of killing the wumpus and let the agent explore further. This gives about 30 more points on average!.
5. If facing a wumpus, shoot it! This also gives about 3 points of improvement.
6. If there is no safe cell to go, do nothing (return Action.NO_OP).
7. If facing the next safe cell decided by the model, move forward.
8. Otherwise turn left or right based on where the next safe cell and nearby wall are located.

The most probable combinations of wumpus and pit locations are calculated by using backtracking search. In order to find the closest next safe cell, a breadth first search algorithm is implemented. Detailed explanations are given as comments in Model.java.

Utils: Defines initializing parameters and gives some utility functions.

Setup:

I commented out some of the print functions to make the simulations run faster.
The following lines in WorldApplication.java have been modified to meet the environment requirements.

line 36: int numTrials = 10000;
line 39: boolean nonDeterministicMode = false;
line 40: boolean randomAgentLoc = false;

Results:

The average score for a 10000 trials run is around 404.79. (see wumpus_out.txt).