Assignment 1

Your assignment is to build an Environment simulator for WumpusWorld, as described below, taken from Russell and Norvig, Artificial Intelligence, A Modern Approach, 3rd Edition:

- Performance measure: +1000 for climbing out of he cave with the gold, -1000 for falling into a pit or being eaten by the Wumpus, -1 for each action taken and -10 for using up the arrow. The game ends either when the Agent dies or when the Agent climbs out of the cave.
- Environment: A 4 x 4 grid of rooms. The Agent always starts in the square labeled [1,1], facing to the right. The locations of the gold and the Wumpus are chosen randomly, with a uniform distribution, from the squares other than the start square. In addition, each square other than the start can be a pit, with probability 0.2.[1]
- Actuators: The Agent can move Forward, TurnLeft by 90 degrees, or TurnRight by 90 degrees. the Agent dies a miserable death if it enters a square containing a pit or a live Wumpus. (It is safe, albeit smelly, to enter a square with a dead Wumpus.)[2] If an Agent tries to move forward and bumps int a wall, then the Agent does not move. The action Grab can be used to pick up the gold if it is in the same square as the Agent. The action Shoot can be used to fire an arrow in a straight line in the direction the Agent is facing. the arrow continues until it either hits (and hence kills) the Wumpus or hits a wall. The Agent has only one arrow, so only the first Shoot action has any effect. Finally, the action Climb, can be used to climb out of the cave, but only from square [1,1].
- Sensors: The Agent has five sensors, each of which gives a single bit of information:
    - In the square containing the Wumpus and in the directly (not diagonally) adjacent squares, the Agent will receive a Stench.
    - In the squares directly adjacent to a pit, the Agent will perceive a Breeze.
    - In the square where the gold is, the Agent will perceive a Glitter.[3]
    - When an Agent walks into a wall it will perceive a Bump.
    - When the Wumpus is killed, it emits a woeful Scream that can be perceived anywhere in the cave.

Instructions

Write a simulator in Python for the Environment as described above. The instructor will post a video that will walk you through an example reference implementation written in Scala and will explain the syntax, how to download and run it, and how it works. The reference implementation is at https://github.com/lasimon/wumpus. For this assignment you want the branch labelled *naïve-agent.*

[1] The location of the Wumpus, gold and pits are independent. The Wumpus, gold and a pit can all be in the same room.

[2] Note that this means that the Wumpus produces a Stench in the square and in vertically and horizontally adjacent squares whether it is dead or alive.

[3] It doesn't matter whether the Agent is holding the gold. If the gold is in the same room, the Agent receives the percept.

(Note that this reference implementation is written in functional style, but you don't need to write yours in the same style. Object-oriented style is also fine for your simulator. Avoid purely procedural style though as it won't scale well for subsequent exercises.)

To test the environment, also write a NaiveAgent that chooses randomly between the six possible actions (Forward, TurnLeft, TurnRight, Shoot, Grab and Climb) with uniform probability.

Some notes and clarifications:

- The Agent must only have access to the Percepts. The Agent should not be allowed to cheat by looking at the game board or by being able to access any other information about the state of the Environment. For example, the Agent CANNOT ask where it is, where the Wumpus is, whether there is a pit in a location, etc. In later assignments we will build Agents that will construct a "mental model" (called a Belief State) of the environment based on the Percepts received so far at each step of the game.
- The Wumpus and pits don't move during a game, but they will move from one game (also called an "episode") to the next.
- We may later want to disallow Climb entirely if the Agent doesn't have the gold. Otherwise the Agent may learn (once we build a learning Agent) to just always leave immediately with a score of -1 given this game is so hard to win.
- When an environment is initialized, provide these parameters:
  - width: width of the grid
  - height: height of the grid
  - allowClimbWithoutGold
  - pitProb: the probability of a pit being added to each square (except (1,1))

  The standard game would be an initialization of (4, 4, true, 0.2). (You can add additional initialization parameters if you wish). We will experiment with larger grids later.

- The Agent does not get any reward for just finding the gold.
- There are all kinds of variations of the above WumpusWorld rules available on the Web. For now, we will stick to the original set of rules as defined in Russell and Norvig (except allowing for variation as allowed by the width, height, allowClimbWithoutGold, and pitProb parameters.) If you try to copy code from the Net it may not be written to use this set of rules so beware.
- The rewards as described by 'Performance Measure" above are to be considered to be granted by the Environment, not by the Agent. Return the reward as part of the percept tuple; do not generate it within the Agent. The reasons for this will become clearer later.
- The squares should be numbered (as far as any user interface is concerned) as [x,y] running from [1,1] in the bottom left corner to [4,4] in the top right. (It's okay if the Environment code internally refers to the bottom left-hand corner of the grid as [0,0] rather than [1,1] to simplify working with arrays.)
- The final Percept, if and when the Agent climbs out of the maze, is not well-defined. The Agent should ignore it, and we could choose something reasonable for it, like (false, false, true if the Agent has the gold, false, false, -1 or 999 depending on whether the Agent has grabbed the

gold). The final reward is very important, but the rest is of no value because the Agent can't take any further actions at this point.

- We could consider the -1 per step as being part of the Agent design rather than an environment reward but for now treat it as being determined by the Environment.
- Include a simple visualizer (such as in the reference implementation) so you can see how the Environment is evolving through the game (despite the fact the Agent can't).
- Hint: When testing, try setting the pit probability to zero to see whether the simulator is working properly if the Agent grabs the gold and escapes. Even the naïve Agent should occasionally succeed.