CSCI 3202 - Intro to Artificial Intelligence
Instructor: Hoenigman
Assignment 7
Due Friday, November 11 by 4pm

Markov Decision Processes

For this assignment, you are provided with Python files, ValueIteration.py and utils.py, that generate an MDP for a given rewards matrix. Your assignment is to modify and run the provided code, including adding additional functionality to the program, to evaluate Value Iteration and Policy Iteration for various inputs.

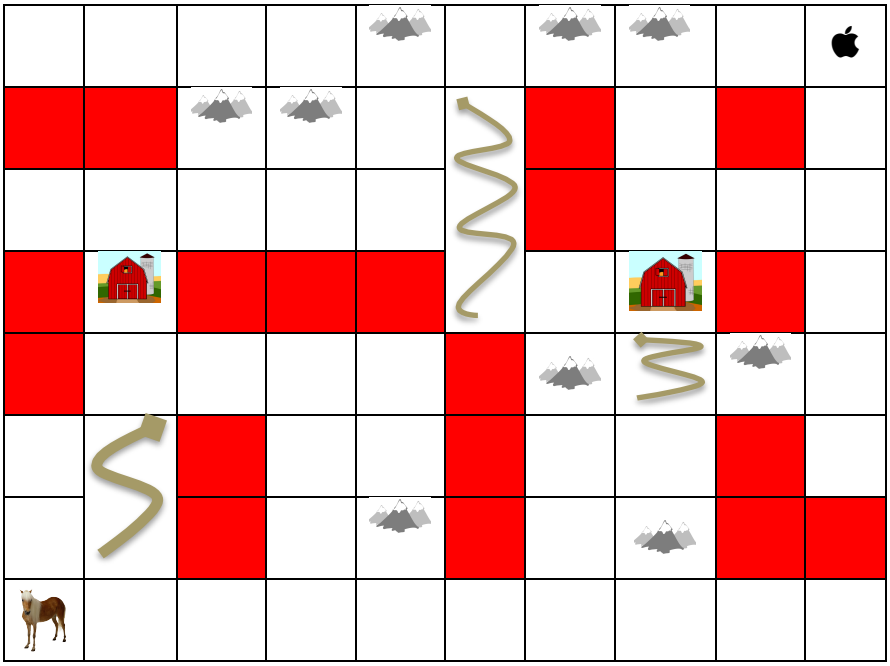
What to submit: Answers to the following questions about how the provided code is working, your revised code that you will need to add for additional functionality in the program, and a short report that describes your results to the questions in the Analyze an MDP problem section.

Questions:
Include answers to the following questions as comments in the code you submit.

1. What action is assigned in the terminal states?
2. Where are the transition probabilities defined in the program, and what are those probabilities?
3. What function needs to be called to run value iteration?
4. When you run *value_iteration* on the MDP provided, the results are stored in a variable called *myMDP*. What is the utility of (0,1), (3, 1), and (2, 2)?
5. How are actions represented, and what are the possible actions for each state in the program?

Analyze an MDP problem:
Once you're comfortable with the provided code, modify the code to run on the following problem.

The image on the next page shows a fictional world where the horse needs to navigate the maze to get to the bushel of apples. Along the way, there are walls that he needs to go around (shown in red), treacherous mountains to navigate, and snakes to avoid. There are also warm and cozy barns where he can recharge. Use the Value Iteration and Policy Iteration algorithms to find the path through the maze that has the highest utility.



World 1. Navigate the maze to get the horse to the apple. Watch out for mountains and snakes.

Rules for the MDP:

- There is an additional negative reward in the mountains of -1.0. The mountains take extra time to navigate and they can be treacherous.
- There is an additional negative reward going past the snakes of -0.5. Snakes bite. Ouch!
- There is a positive reward of 2.0 if the horse gets to a barn.
- The horse can't move through a wall.
- The horse can move in four directions only: left, right, up, down.
- The reward for getting the apples is 50.0.

Use the transition model already implemented in the program, where the horse is successful 80% of the time, 10% of the time he will go left of his intended move and 10% of the time he will go right of his intended move.

Experiments

Modify the code as necessary to evaluate the impact of the following variables on the results:

1. Living reward.
    a. For all non-terminal states that are not mountains, barns, or snakes, are there any values between -1 and +1 that change the results of the program?
    b. Are there any values that result in the program not finding a path to the terminal state?
2. The value of γ.

a. Are there any values between .1 and 1 that change the results of the program?

Another code modification

What if the horse could jump? Once you've implemented changes to evaluate the living reward and gamma on the results, change the actions in the program to include jumping, where the horse can jump over a state in any of the left, right, up, down directions. For example, if the horse is in state (0, 0) and jumps up, he will land in state (0, 2). Part of the jumping action needs to include evaluating whether the jump lands the horse in a valid state. If there is a wall at (0, 2), then the jump is not a valid action. Because the horse is a horse, he's not good at jumping and only succeeds 50% of the time. The other 50% of the time, he stays in the same state. You will need to change the program to accept a different transition model than the one currently implemented.

The code modification you make to evaluate the living reward, gamma, and the transition model should be implemented as arguments to the necessary functions so that the code can be easily run.

What to include in your report

Your report should have the following sections:

Purpose: Describe the purpose of the assignment and what you want to accomplish.

Procedure: Your procedure section needs to include the modifications you made to the code and the experiments you ran, including the range of values used in each experiment.

Data: Describe the data, including specific values in the matrix and what they represent.

Results: Describe the results of the experiments modifying the living reward, gamma, actions, and transition probabilities for the jumping action.