Artificial Intelligence
Project 3

Overview

The goal is to help Pacman navigate a game board and survive the ghosts. The game is a little different then the arcade version:

- There is a fixed amount of time your agent has to decide on its move.
- The ghosts always move closer to you (except when they're scared)
- You get 10 points for eating each food pellet
- Every turn you lose 1 point, so you need to be efficient
- Eating a blue power up gives you 40 turns where the ghosts are scared. You get 200 points for each on you catch. During this time they move away from you at half speed.
- If you clear all of the food pellets you get 500 points and the game ends
- If a ghost catches you, then you lose 500 points and the game ends.

If you have cs1graphics, you can watch everything happen using a graphics screen. Otherwise, you can see the game board in text form.

Files: The files are available in the class git repo (and also on Moodle). The the course page for instructions on access the git server. The name of this assignment is `project3`. You should only edit MyPlayer.py and description.txt.

Grading: Each assignment will be graded as follows:

50%	Performance on final version relative to reference implementation
30%	Code quality: correctness, use of AI methods, clarity, etc. this will be based on both the initial and final version
20%	Description of algorithm used, why it is effective and the steps to took to improve it
up to 5% bonus	Performance relative to your classmates

Submission

Due: Final version 6am Wednesday, April 8

Goal: Write an agent that scores as high as possible in this Pacman variant

Testing: You can test your program using `Run.py`, see below.

Submission: Perform a git commit and a git push to submit your files. Submitting your files will automatically test your code and post results on the contest page. This way you know how you perform relative to your classmates and the reference implementations. Note that the final grading will be using a different set of test instances. Programs will be given a maximum time of 2 minutes to solve each puzzle.

Late penalty: Projects will receive a 5% penalty for every day that the initial versions are late. Submissions more than 3 days late will not receive detailed comments for resubmission. Final version must be submitted on time.

Grading Note that a significant portion of your grade will come this version. If you submit your project early, you can get feedback and improve it further.

Feedback and resubmission

When you are ready to submit, you should push your project to the git repository and email me with the git commit id (a hexadecimal key located on the list of files in your repository). I will provide feedback in the order received. I will try to get feedback to you as quickly as possible. However, there are a lot of projects to review. I will guaranteed a response withing 72 hours. If you choose, you can refine and improve your code and project description.

What to change in MyPlayer.py

You should carefully go through the code in MyPlayer, you don't have to worry about the code in the other files. You will be implementing the `findMove` methods. You can add any helper functions you want and make additions to the constructor.

Running your program

In either the Python shell or Idle run `Run.py`. It will ask you to enter a few parameters and then will run.

Useful methods

Player class (which MyPlayer inherits from):

- `timeRemaining()` return True or False depending on whether there is time remaining to determine your move
- `setMove(move)` sets the move you want to use. You can call this multiple times as you find better move options. If time is up, this does nothing.

State class:

- `getScaredTurnsLeft()` returns how many turns are left for the ghosts being scared. 0 if they are not.
- `getPlayerPosition()` returns (row,col) for the position of the players
- `getGhostPositions()` return a list of (row,col) positions for all of the ghosts
- `getPellets()` returns a set of the locations (row,col) of all of the pellets
- `getPowerUps()` returns a set of the locations (row,col) of all of the power ups
- `gameOver()` returns true if the game is over
- `getScore()` returns the score
- `getTurn()` the current turn of the game
- `actions()` return a list of the possible actions the player can make from it's current positions. Possibilities are 'Stay', 'Left', 'Right', 'Up', 'Down'.
- `result(action)` returns a new state that is the result of the player taking the given action
- `ghostResultDistribution()` return a list of (state, probability) pairs. These include all possibilities of where the ghosts may have moved.

Hints

To do well consider the following:

- Submit agents often. Test them, find improvements, document and submit again.
-