CS 221 – Autumn 2018
Anand Natu
anatu - 06264867

Homework #5: Pac-Man

Problem 1

Problem 1a

For this problem, we have n+1 players consisting of a single agent (Pacman) and the n opposing players (ghosts). In order to understand how to write down our value function, we can express each state as follows:

- **At the end state**, the result is simply the utility derived from that state as determined by the game's utility function.
- **At the maximum depth**, the agent uses the evaluation function for the maximizing selection instead of searching further into the tree
- **If the player is the agent,** the value function will be recursively called for all of the possible successor states, and the maximum of these will be selected (if the maximum depth has not yet been reached)
- **If the player is an opponent, but NOT the last opponent,** the value function will be recursed and the minimum will be taken (since all opponents are min agents)
- **If the player is the LAST opponent,** the value function will be recursed and the depth decremented by one (since a single depth layer constitutes the movement of the agent and ALL of the opponents)

Therefore, the recurrence of the value function can be written as:

$$V_{minimax}(s, d) = \begin{cases} Utility(s) & isEnd(s) \\ Eval(s) & d = 0 \\ \max_{a \in Actions(s)} V_{minimax}(Succ(s, a), d) & Player(s) = agent \\ \min_{a \in Actions(s)} V_{minimax}(Succ(s, a), d) & Player(s) = opponent_i, i < N_{opps} \\ \min_{a \in Actions(s)} V_{minimax}(Succ(s, a), d - 1) & Player(s) = opponent_i, i = N_{opps} \end{cases}$$

CS 221 – Autumn 2018
Anand Natu
anatu - 06264867

Problem 3

Problem 3a

The expectimax function should be identical to the minimax function defined for Problem 1 in all respects except for the behavior of the ghost players, since their behavior is being changed from a minimizing behavior to the expected value of their behavior based on the random policy π_{rand}. Therefore, the expectimax recurrence is given as:

$$V_{expmax}(s,d) = \begin{cases} \textbf{Utility}(s) & \textbf{isEnd}(s) \\ \textbf{Eval}(s) & d = 0 \\ \max_{a \in Actions(s)} V_{expmax}(Succ(s,a),d) & Player(s) = agent \\ \sum_{a \in Actions} \pi_{rand}(s,a) V_{expmax}(Succ(s,a),d) & Player(s) = opponent_i, i < N_{opps} \\ \sum_{a \in Actions} \pi_{rand}(s,a) V_{expmax}(Succ(s,a),d-1) & Player(s) = opponent_i, i = N_{opps} \end{cases}$$

Problem 4

Problem 4b

At a high level, I wanted my evaluation function to "augment" the base score of the given state s (i.e. the default evaluation function used for the other parts of this assignment) with additional information. Therefore, the calculation begins with just taking the base score and adding in the following information:

1. The total distance between Pac-Man and all of the other ghosts on the board (higher is better)
2. The distance between Pac-Man and the nearest ghost (higher is better)
3. The distance between Pac-Man and the nearest food pellet (lower is better)
4. The distance between Pac-Man and the nearest capsule (lower is better)
5. The behavior of the ghosts – if they are not scared, then the score is de-rated depending on how close they are to Pac-Man. Conversely, if they are scared, then the score is up-rated if they are closer to Pac-Man.

For items 1-4 above, reciprocal scores are used, while for item 5 a simple score addition / subtraction was used as a function of the distance between Pac-Man and the ghosts. No other external weights were applied when adding these features together.

Finally, the score is de-rated at the end to account for any outstanding food pellets and capsules that are still left on the board. These weights were decided somewhat arbitrarily, but were made to be less than 1 in order to be generally consistent with the modifications imposed by the reciprocal features described above.