Pacman reflex agent

# Reflex Agent

Reflex agent that evaluates all possible action from the current state and gives a score to that action based on how good that action is

# initial value calculation

In every turn our agent does some initial calculations and makes use of dictionaries to make the new calculated information available in the whole class. Most values are calculated for the initial game state and the state after each of the legal actions pacman can make.

self.pacmanPositions contains pacmans position before and after each legal action

self.ghostDistances contains all the maze distances between pacman and the ghosts, before and after an action of pacman.

self.foodDistances contains all the maze distances between pacman and all food pellets, before and after an action of pacman.

self.ghostSpawns contains all the positions where ghosts spawn and next to where the ghosts spawn. Our agent uses this information to determine if he can eat a ghost without the ghost spawning on top of pacman.

self.ghostDangerValues contains a value for each ghost on how dangerous he is for pacman. This value is calculated for each ghost for each pacman action

# Score calculation

Our agent gives a value for each action pacman can take and will take the action that has the highest score. The value for each action is calculated separately and are relative to the initial position of pacman. There are two main factors in giving a value. The first is based on a score for food. The second is based on how dangerous a tile is for pacman to be caught by a ghost.

The score for food is calculated based on the maze distance to the closest food pellet before taking an action in comparison to the maze distance to the closest food pellet after an action. This means that each action gets a positive value if it takes pacman closer to a food pellet and negative if pacman gets further from a food pellet. When there is no difference the food score will be zero.

The score for ghost is calculated for each ghost individually. This means that we have a ghost dangerousness value for each ghost per action pacman can take. The score is based on how dangerous the ghost is for pacman. Actions that lead to a maze distance less than two between pacman and a non-scared ghost are considered very dangerous, because such an action leads to death. Because death situations should be avoided at all cost we give this action a negative score. Only when pacman moves towards a scared ghost the ghost danger value for that ghost will be positive. We only consider moving towards a scared ghost good and don’t consider moving away bad. This is to prevent conflict between real bad actions and actions that may seem bad. We consider survival first priority and optimization second.

Our agent did die in some (at first) unexpected ways. When our agent ate scared ghosts these ghosts would respawn. In some situations these ghosts spawned on the same place or next to pacman. This meant that after eating a ghost, pacman was instantly killed by the ghost pacman had eaten. We solved this by storing the positions where a ghost can spawn and the positions next to these positions for each ghost. We used these positions to check if we can safely eat a scared ghost. If a ghost is walking on his spawn area pacman waits until the ghost leaves this area before eating the ghost.