

question 1:simple evaluation function giving very very bad "evaluation on moves at positions where ghosts are close than 3 positions

and favoring moves to positions closer to food

questions 2,3,4: simple implementations of the algorithms minimax, alpha_beta, expectimax searches(max player pacman, min player ghosts)

question 5: this will be an evaluation function based on distances

the highest priority for pacman will be not being eaten by ghosts while simultaneously eating as

many dots as possible (however eating for this function is not more important than avoiding ghosts)

note that being too far from ghosts is meaningless as long as you have safe distance from them (lets say 5)

(thats why we take the max between 5 and distance_from_ghost in order not to favor positions that ghosts are already too far and are not being a threat for pacman while favoring the safe distance 5)

so we have this \rightarrow score += max(Dist_from_Ghost, 5) * 2 (multiplying by 2 makes the argument really strong because as we said the most important thing is not being eaten)

so now as we have strongly kept pacman away from ghosts we need to favor positions-moves that will make pacman eat as many dots as possible

so we have this \rightarrow score -= minDistFromDot * 2 i subtract from the score the distance to the closest food multiplied by 2 in order to get closer to the food in every move

This, however, i need to add a penalty for the remaining food because the above practice make the pacman be avoid secluded food (food in a distant area)

so \rightarrow score -= 3.5 * len(dot_positions)(bonus if you minimize the number of the remaining food in general)

lastly,

i believe that we need to favor pacman to eat the ghosts and add a bonus in eating capsules(subtract a value*its distance from capsules)

the value will be 4 to push pacman slightly being closer to capsules

so \rightarrow score -= 4 * len(capsule_positions)