HW2 Documentation
Introduction to Artificial Intelligence and Machine Learning

October 25, 2017

Professor Tian-Li Yu

電機四 B03901018 楊程皓

Problem 5

My evaluation function considers multiple distances situation: the win state, the lose state, the distance of the nearest food, the nearest ghost in a self-defined dangerous range, and the nearest scared ghost if reachable, the nearest capsule and possible reachable ghosts after eating it. With each elements are weighted with a reasonable constant will result in a good enough evaluation function that passes the tests and have about 1615 points in average over 120 games.

With a simple BFS, I get all the real shortest distances to every grid in the map from the current agent, in order to prevent stuck in a place if using manhattan distances. The weight constants are as follow:

1. Win/Lose: 1000, if the state wins or loses.
2. DangerousWeight: 600, if the nearest non-scared ghost is within a dangerous distance (safeDis defined as = min(3, width - 2, height - 2)). The evaluation function is weighted by (safeDis - dis) * dangerousWeight / safeDis.
3. expEatWeight: 150, if the nearest ghost is reachable before the scaredTime, considering the agent is 2 times faster than a scared ghost. The weight is set as 200 because the bonus to eat a scared ghost is 200.
4. eatFoodWeight: 10, calculating the nearest food's distance.
5. Capsule: calculating the nearest capsule and the nearest ghost assuming the expected distance remain the same since the ghost choose steps uniformly random. The evaluation function is weighted by expEatWeight / 2 * (maxScaredTime - 2 minDis - 2 minDisToGhost) / maxScaredTime, where maxScaredTime = 40.

With the weights described above, the agent behaves reasonably, considering win/lose first, then dangerous situation, potentials to eat ghosts (with or future capsules), and the nearest food last considered.