

1. ReflexAgent

The evaluation function takes in the current and proposed successor game states and returns a number (higher the number, the better it is)

In the evaluation function, the score for successor state is calculated in:

‘evaluatedScore’ which takes in account the total scare time, food count, max distance, capsule score and successor game state score.

- Higher the total scare time for a successor, the better it is.
- The successors actions=”Stop” should not be allowed.
- The pacman and ghost aren’t allowed to be at the same position.
- The evaluation score must be increased if the successor game state leads to a win.
- The successor state with the minimum distance to the food is chosen.
- If the number of food present in the successor state is less than that in the current state, increase food count by 100 for that successor state.
- Increase the capsule score by 100 if pacman eats a capsule.

2. Minimax

Minimax uses the follow functions:

- Value: If the state is a terminal state, then return the state’s utility. If the next state is MAX, then return its value. If the next state is MIN, then return its value.
- Max-value: for each successor of state, maximize the value of the max node of that state.
- Min-value: for each successor of that state, minimize the value of the min node of that state.
- Moves: choose the best move to be executed by pacman.

3. Alpha-Beta Pruning

Alpha-beta pruning uses the following functions:

- Value: If the state is a terminal state, then return the state’s utility. If the next state is MAX, then return its value. If the next state is MIN, then return its value.
- Max-value: (alpha)Max’s best option on path to root.
- Min-value: (beta)Min’s best option on path to root.

4. Expectimax

Same as minimax except that ‘expctvalue()’ is used in replacement of ‘value()’ and ‘min()’

The expctvalue() calculates the probability of the successor and multiplies it with evaluation function value of each successor node.