

# COMP7404 Assignment 2 Short Report

SHI Zhongqi 3035347861

## 1 Question 1

Use the gameboard score as the base score. If the pacman didn't move or next to the ghost, it would get minimal score.

The other part of the score is distance to the nearest food. Here is a minus to make the pacman get highest score when it approached a food.

## 2 Question 2

Used the recursion to complete the minmax function.

The depth will only be increased when the agentID equals to the total number of agent in the game.

Then use agentID to determine whether it need to do the max or min.

Still "stop" is not welcomed in the design.

## 3 Question 3

Just like the pseudocode, add the variables, alpha and beta to the min-max function.

When do the max-value selection, if the game state score is bigger than beta, then return the score. If score not bigger than the beta but bigger than the alpha, then let alpha equals to the score.

When do the min-value selection, if the game state score is smaller than alpha, then return the score. If score not smaller than the alpha but smaller than beta, then let the beta equals to the score.

## 4 Question 4

Just like the pseudocode, make some changes to the min-max function.

When ghost moves, the score will not be compared with each other.

It will be add together and returns an average value.

```
else:
    #minValue=float('inf')
    avgValue=0
    indicator=0
    for action in gameState.getLegalActions(agentID):
        if action == "Stop":
            continue
        successor=gameState.generateSuccessor(agentID, action)
        tmpValue=self.minMax(agentID+1,successor,depth)
        indicator+=1
        avgValue+=tmpValue[0]

    avgValue=avgValue/indicator
    return (avgValue,0)
```

## 5 Question 5

The Score are consist of:

1. BaseScore of the CurrentBoard
2. Distance to nearest Food
3. Distance to the ghost \* 5
4. Distance to the Scared Ghost

To get the highest score, the pacman should not only eat food but also eat scared ghost.

To eat the scared ghost, it will not be too scare of the ghost.

Thus, the distance to the ghost will be counted in the mechanism.

I have tried to times the distance from 2 to 10, it performs best when it is 5.

Therefore, here the distance to the ghost will times 5.

## 6 Question 6

The key is the P-position and fingerprint.

Mark their coordinate and put them in a list is the easiest way to build the function of selection. Since C and CC is a bit special, so the judgement is done by the If and else condition.

```
#fingerPrint
stateA = [(0,8),(1,3),(1,7),(0,1,6),(0,2,4),(0,2,7),(0,4,5),(0,1,3,4),(0,1,3,5),(0,1,3,8),(0,1,7,8),(0,2,6,8),(1,3,5,7),(0,1,4,5,6),(0,1,5,6,7),(0,1,5,6,8),(0,1,3,5,7,8)]
stateB = [(0,2),(0,4),(0,5),(1,4),(0,1,3),(1,3,5),(0,1,4,5),(0,1,4,6),(0,1,5,6),(0,1,6,7),(0,1,6,8),(0,2,4,7),(0,4,5,7),(0,1,3,5,8),(0,1,3,5,7)]
stateC = [(0,1,5),(0,1,7),(0,1,8)]
stateD = [(0,1,4),(0,2,6),(1,3,4),(0,1,5,7),(0,1,5,8),]
stateAD = [(0,1)]
self.stateDict["A"] = stateA
self.stateDict["B"] = stateB
self.stateDict["C"] = stateC
self.stateDict["AD"] = stateD
self.stateDict["AD"] = stateAD
self.winList = ["CC","CB","BC","BB","A"]
```

In order to avoid the complicated status of different result for the timing, the string is used like “CC”, “CB”, “BC”, “BB”, “A”.

Then another difficult part is to do the mirror and turning for the fingerprint. Because it is really hard to mark all the finger print. So, I choose to do the computation.

```
#turn 90 180 270 degrees
for i in range(3):
    tmpBoard = self.turnDegreeBoard(board,i+1)
    for m in range(9):
        if tmpBoard[m]:
            xPositions[i+1].append(m)
#0,1,2,3
#Mirror of the bord
mirrorBoard = self.turnMirrorBoard(board)
for i in range(9):
    if mirrorBoard[i]:
        xPositions[4].append(i)
#turn 90 180 270 degrees for mirror
for i in range(3):
    tmpBoard = self.turnDegreeBoard(mirrorBoard,i+1)
    for m in range(9):
        if tmpBoard[m]:
            xPositions[i+5].append(m)
```

First to get all the board status results, then check the actions, if the action could lead the board result to one of the P-Postion, then return it.