COEN 266 Artificial Intelligence

Homework #5: Pacman-II

Name:

Benjamin Wang

Christopher Tam

Himanshu Dahiya

ID:

1179478

1102598

1629215

Contents

[Task: Minimax Agent (Code) 2](#_Toc96963442)

[Comment: 3](#_Toc96963443)

[Contribution Percentage 5](#_Toc96963444)

# Task: Minimax Agent (Code)



Figure - Part 1 of 2 of Code



Figure - Part 2 of 2 of Code

# Comment:



This minimax\_value\_func() corresponds to both the maxValue\_func() and minValue\_func() example sub-functions in the HW5\_submission\_sample PDF as it does the function of both.



Since we’re recursively calling minimax\_value\_func(), we define the exit conditions at the beginning of this sub-function as when the node is a leaf aka terminal state or has reached the set depth just return the value.



This section defines the available branches of a node, the successor’s agent Index (playerIndex), and keep track of the successor’s depth. The logic for successor’s depth is that if its playerIndex resets to 0, it means we have gone through 1 ply/depth so add 1 to the tracker (currentDepth), otherwise stay same.



This is where the recursive call to itself takes place with each call returning only the best value with the exception for the first call from player 0 (Pacman) at depth 0 as will be explained later. This code snippet and the previous code snippet can be condensed into a one-liner if the previous code snippet’s three variables (legalMoves, succPlayerIndex, and succDepth) are inserted into where they are used in this code snippet, but we chose to not do this for better readability.



Best value or best score is defined as the maximum and minimum of recursively returned values if the agent is Pacman (agent index == 0) and if the agent is Ghost (agent index >= 1), respectively. This is the literal only difference between a maxValue\_func() and minValue\_func() and why they shouldn’t be split into two functions.



Ignoring the if-case for a moment, all but one special case calls for the minimax\_value\_func() to return only the value. If and only if the current depth is at 0 and the agent is Pacman, meaning we’re at the very start/top of the game tree, then we want to also return the best action. We first get the indices of the best scoring branches, of which there might be several with exact same scores, then pick one among them fairly by random choice, then finally use that index to map it to the corresponding legal move as best action.



This is the part of the getAction() that initiate the recursive call to minimax\_value\_func(). We know that we always start at Pacman, aka agent index 0, and we start the tracker for depth at 0. Root\_node\_value or the best\_score is returned so we can print it out, and we return the calculated best\_action without updating the gameState because this is just returning what action to take, not taking the action itself.

# Contribution Percentage

Benjamin Wang 40%

Christopher Tam 30%

Himanshu Dahiya 30%